Distributed Ray Tracing

- Maj. Mike Villandre
  - Lecture Method
- Stolen Slides and Pictures

References

1. Robert L. Cook, Distributed Ray Tracing, Proceedings of SIGGRAPH 1984; pgs 137-145.

2. John D. Hart, Distributed Ray Tracing CS319 Advanced Topics in Computer Graphics; 2001; www-courses.cs.uiuc.edu/~cs319/

3. www.siggraph.org/education/materials/HyperGraph/toc.htm
   - Radiance Paper (SIGGRAPH ’94)
     - Introduction
     - 2.3 Support a Variety of Reflectance Models
     - 3.3 Adaptive Sampling of Light Sources

Why Ray Tracing Looks Fake

- Jagged edges
- Hard shadows
- Everything in focus
- Objects completely still
- Surfaces perfectly shiny
- Glass perfectly clear

- Solution: Distributed Ray Tracing

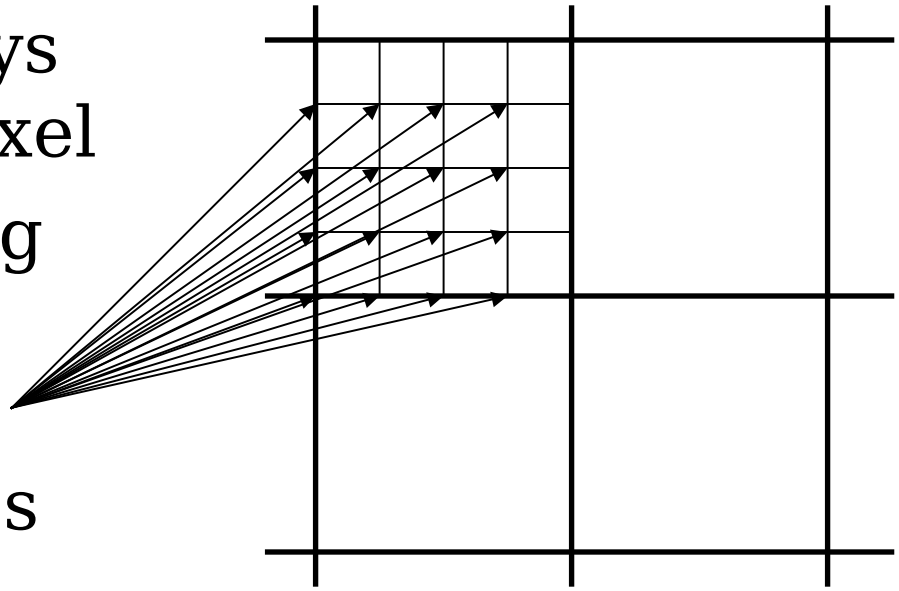
Distributed Ray Tracing

- Replace single ray with distribution of rays
- Rob Cook, SIGGRAPH 84

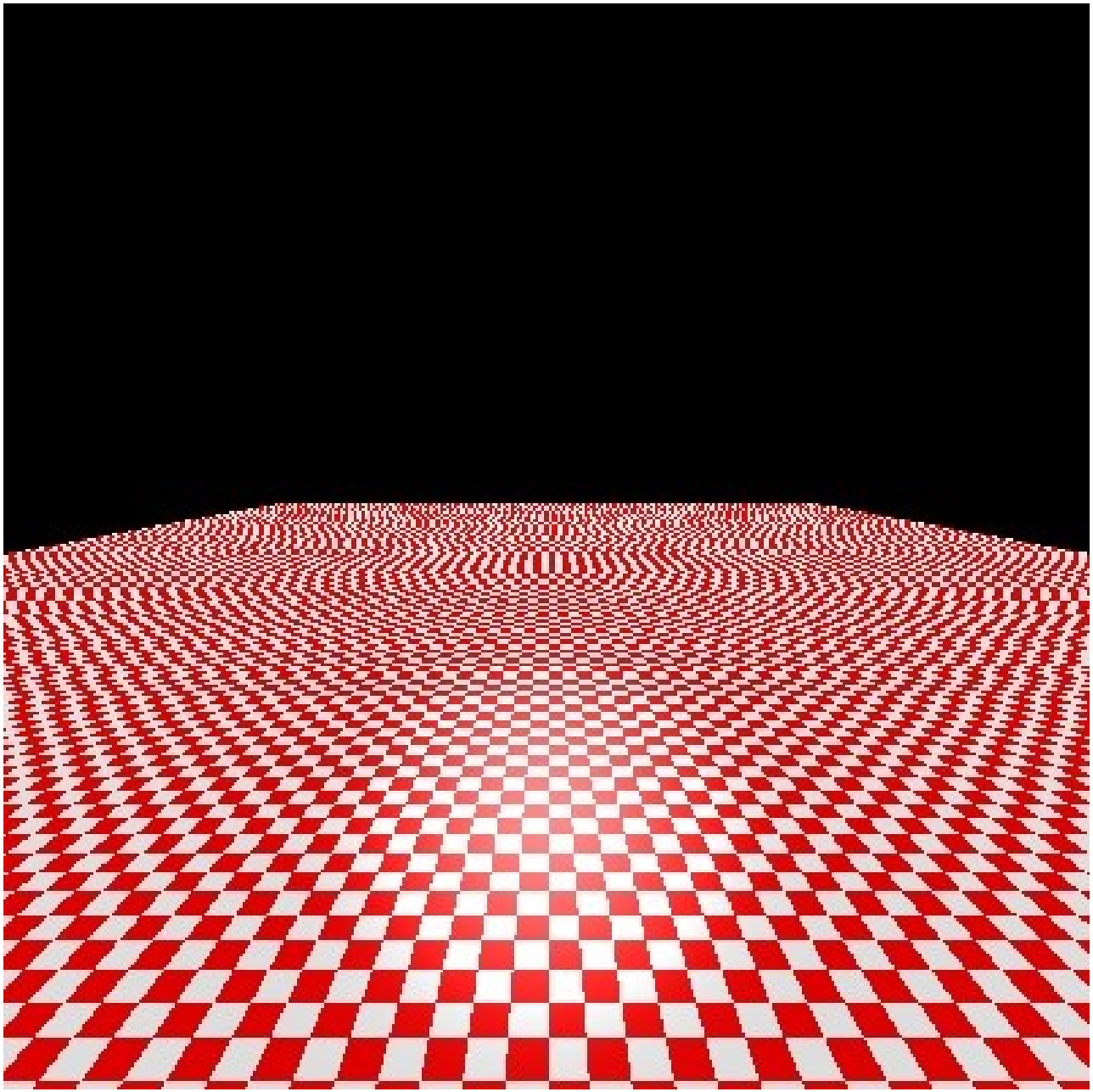
- Multiple eye rays
  - anti-aliasing
  - motion blur
  - depth of field
- Multiple shadow rays
  - soft shadows
- Multiple reflection rays
  - glossy surfaces
- Multiple refraction rays
  - diffuse glass

Supersampling

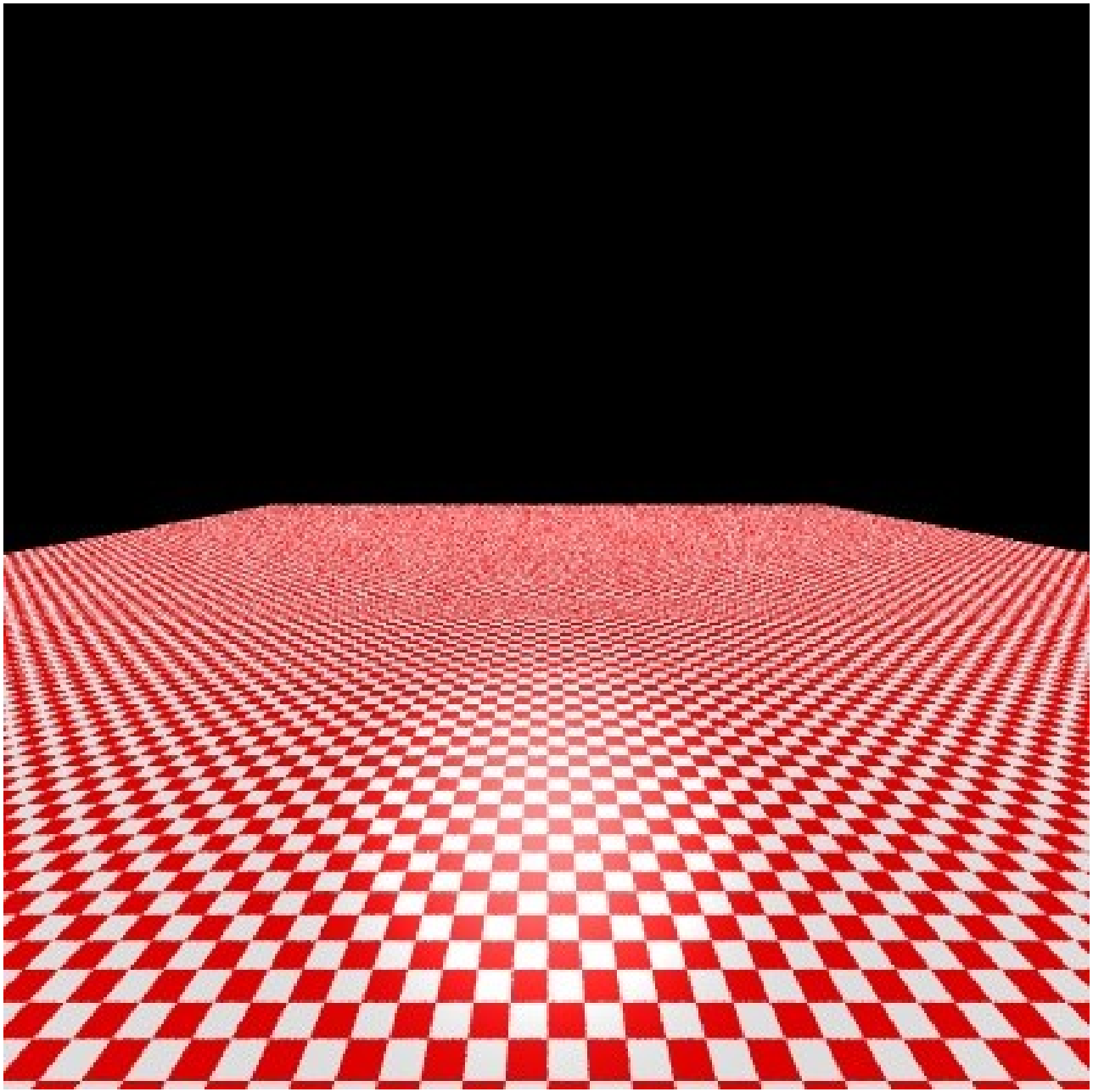
- Cast multiple rays through same pixel
- Average resulting color during reconstruction
- Jittering samples disguises aliases as uncorrelated noise



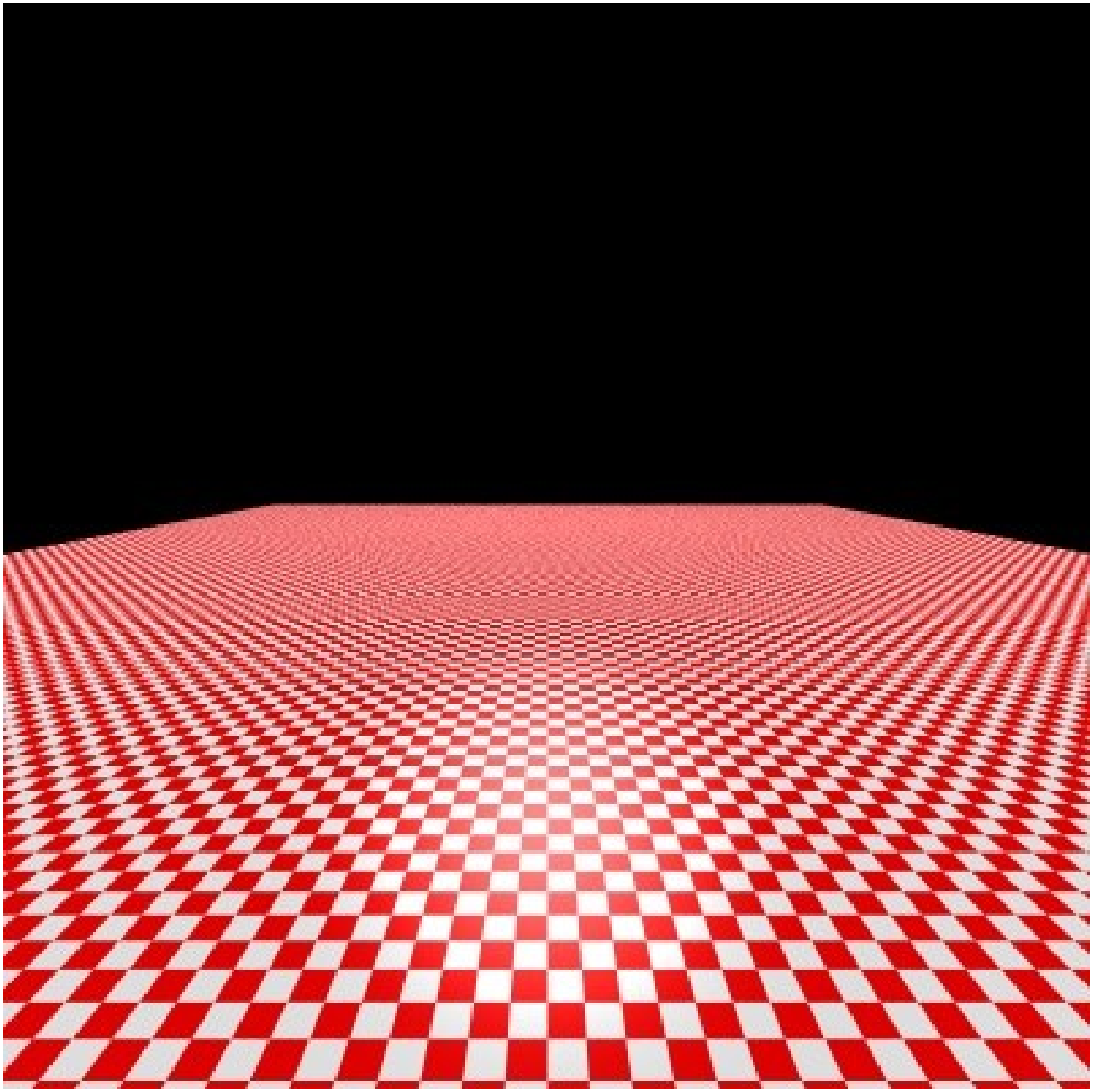
$$I(x,y) = r(x,y) * \int\int f(x,y)s(x,y)dxdy$$



1 sample per pixel, 5 seconds to generate



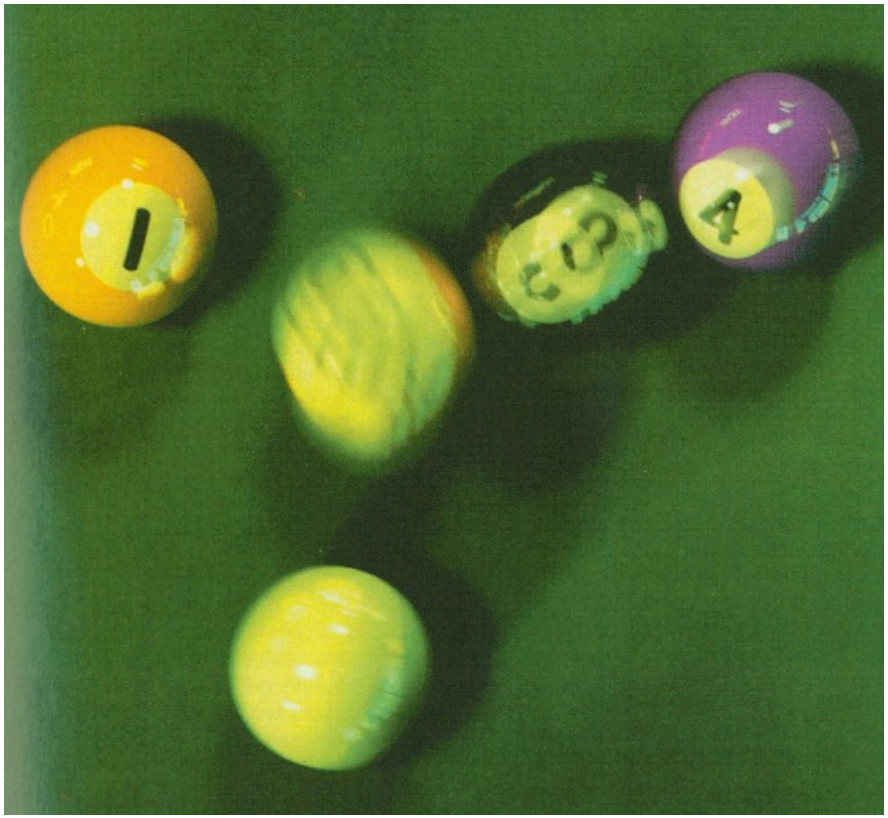
16 samples per pixel, 32 seconds to generate



256 samples per pixel, 571 seconds to generate

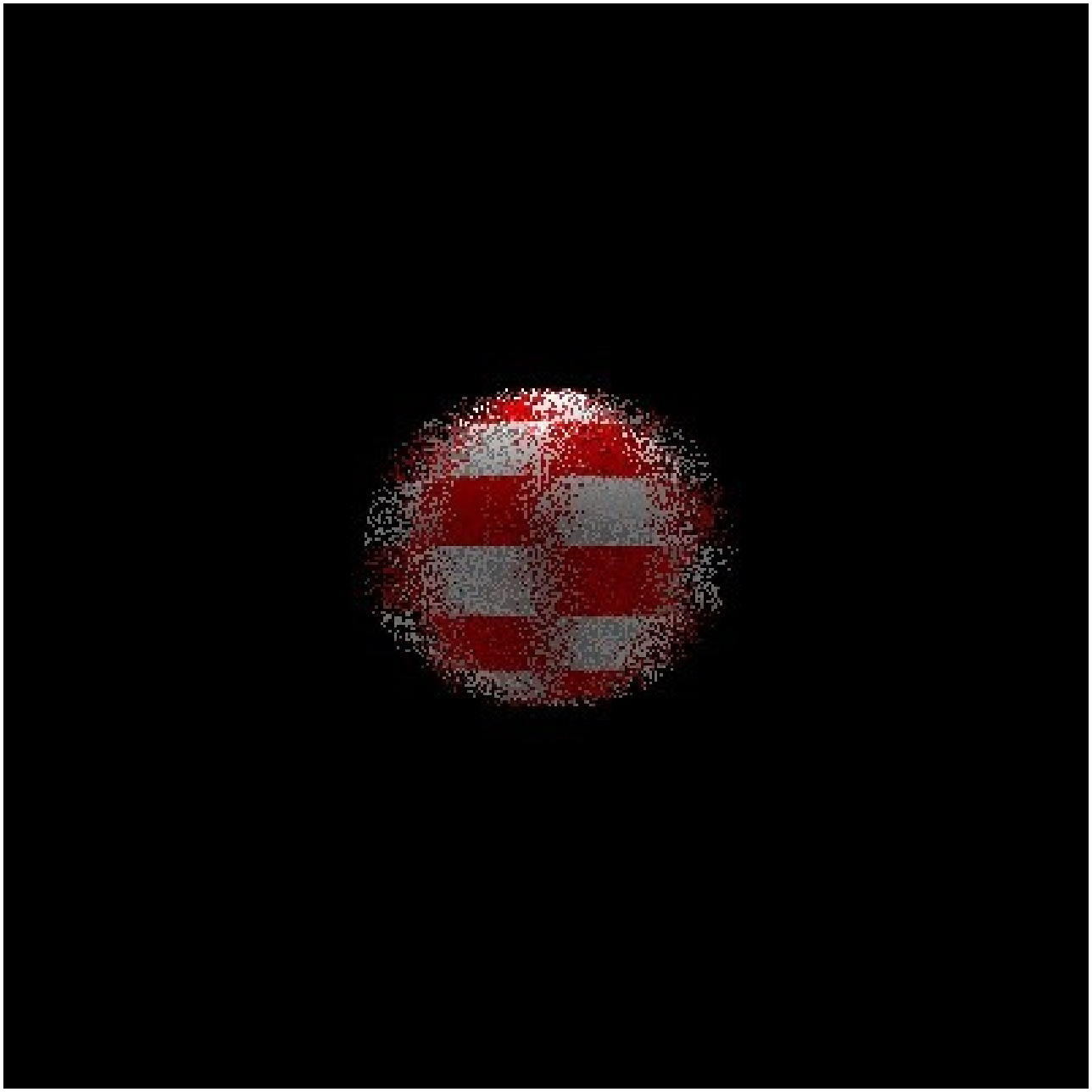
Motion Blur

- Cast multiple rays through each pixel
- Intersect each ray with scene at a different time
- Reconstruct samples with appropriate filter
  - Box filter – fast shutter
  - Triangle filter – slow shutter

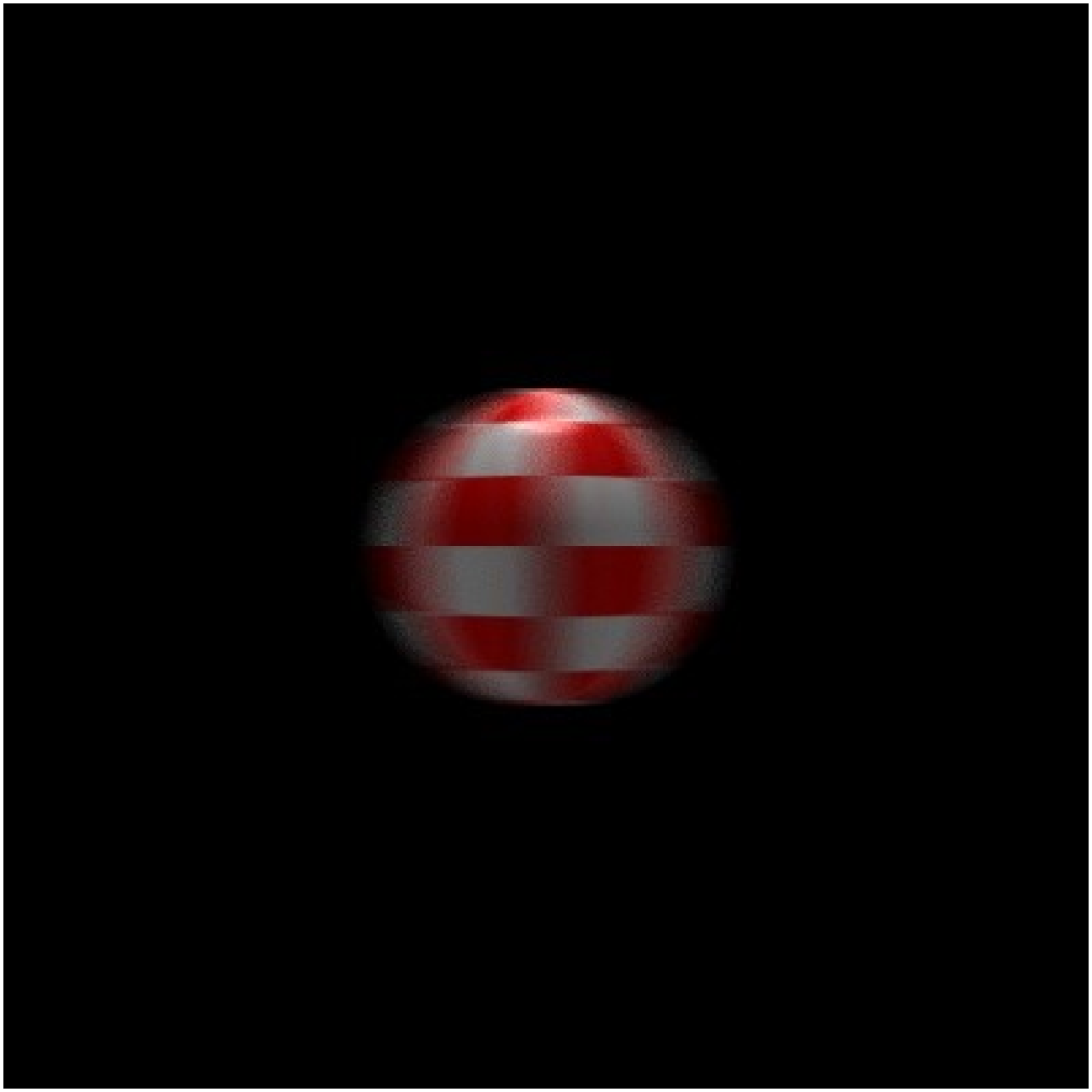


$$I(x,y) = r(x,y) * \int f(x,y,t)s(x,y,t)dt$$

At Rest

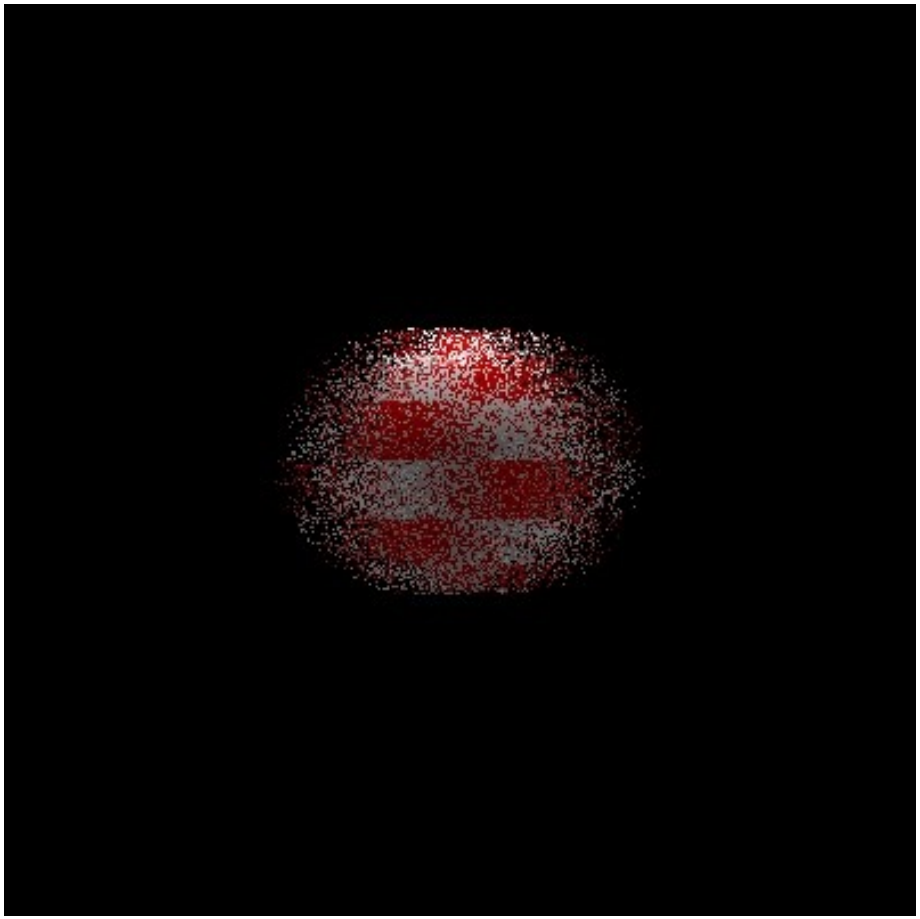


1 sample per pixel

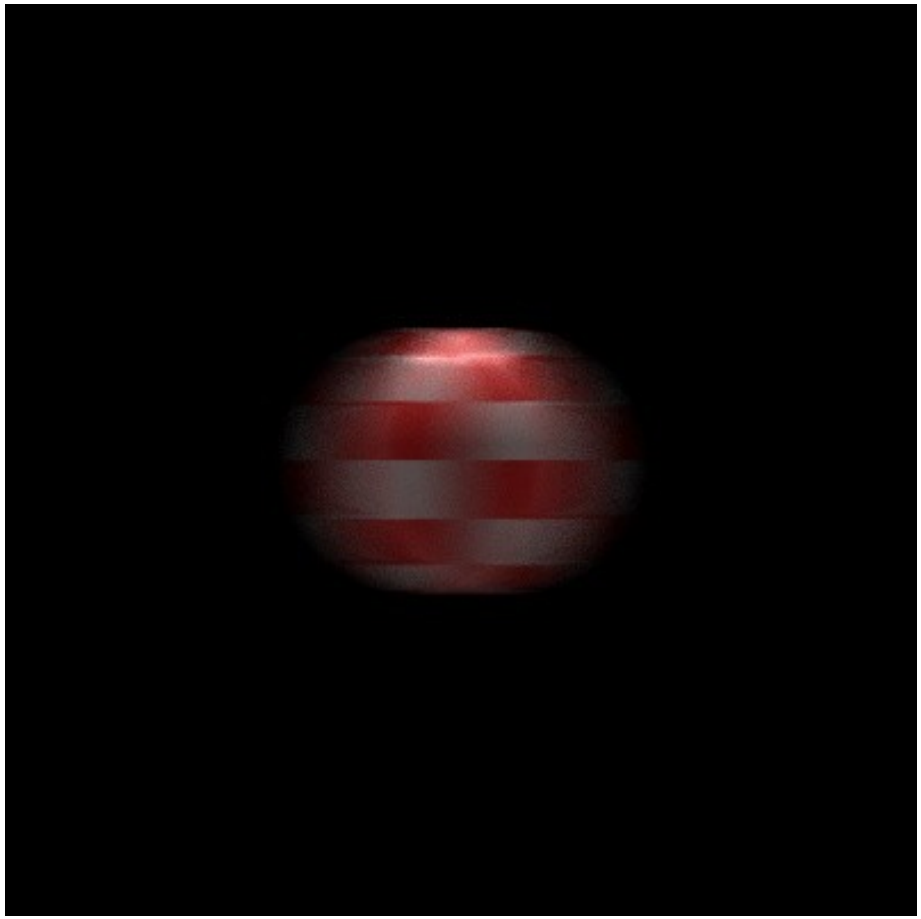


256 samples per pixel

Motion Blur



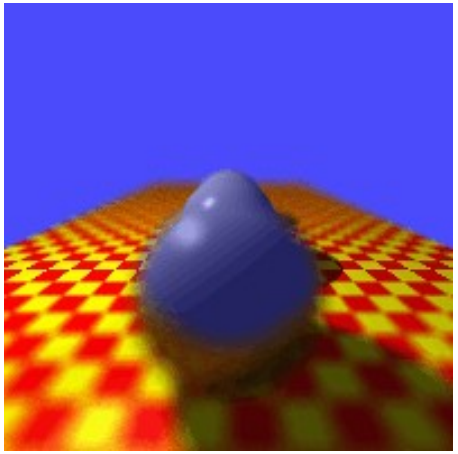
1 sample per pixel

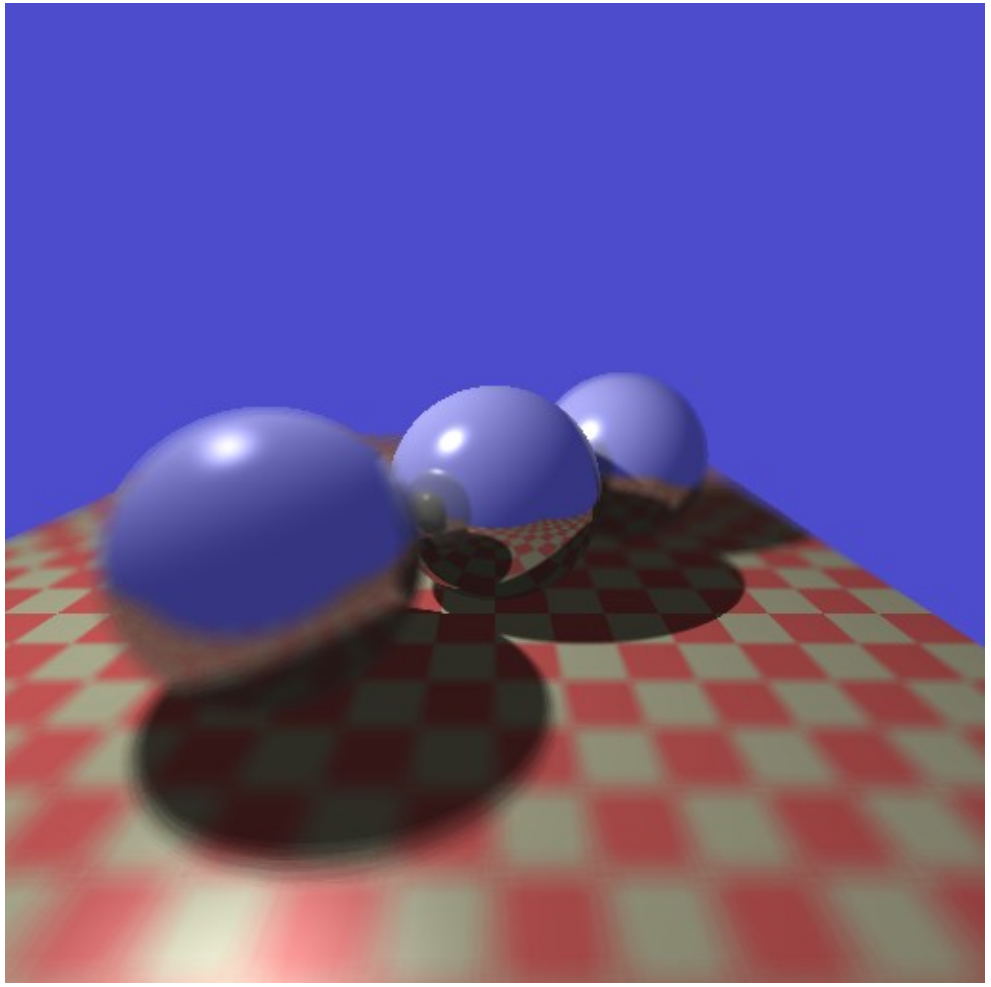


256 samples per pixel

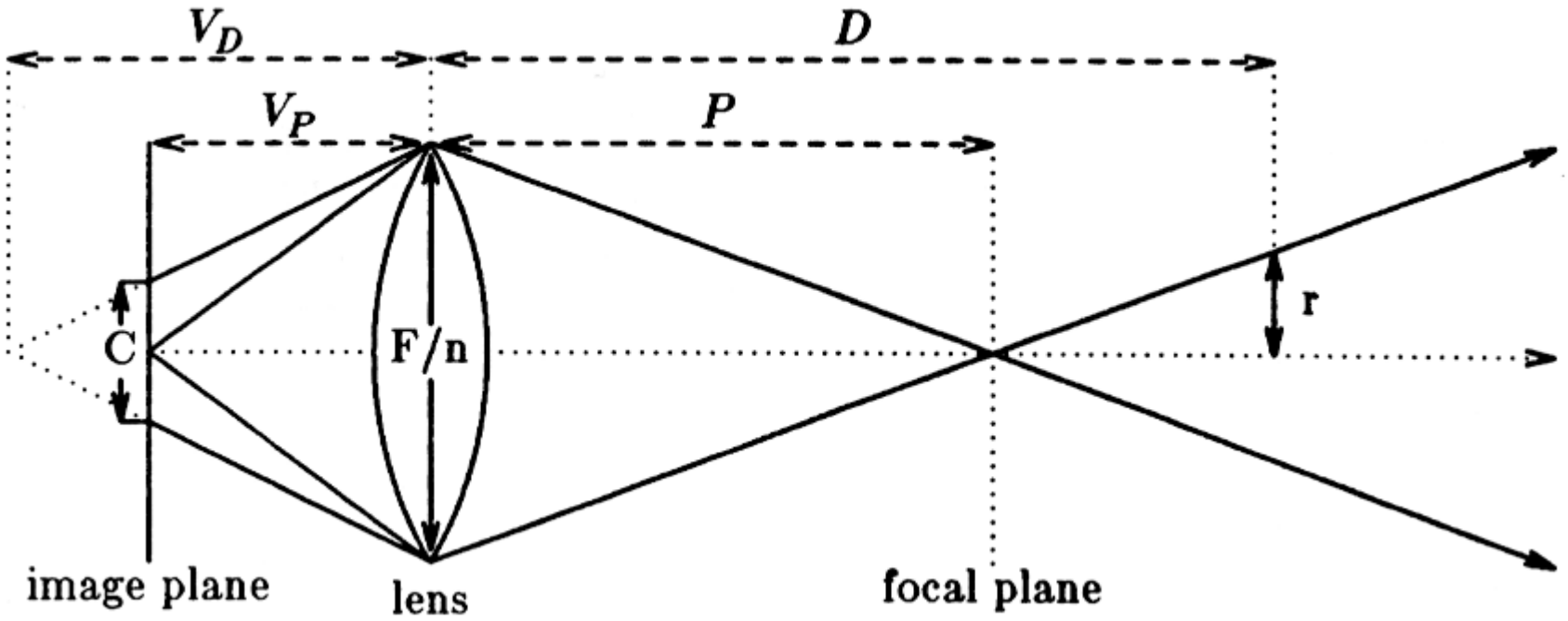
Depth of Field

- Better simulation of camera model
  - f-stop





Camera Model



F – focal length

n – aperture number

C – circle of confusion

$V_P = FP/(P\text{-}F)$

$V_D = FD/(D\text{-}F)$

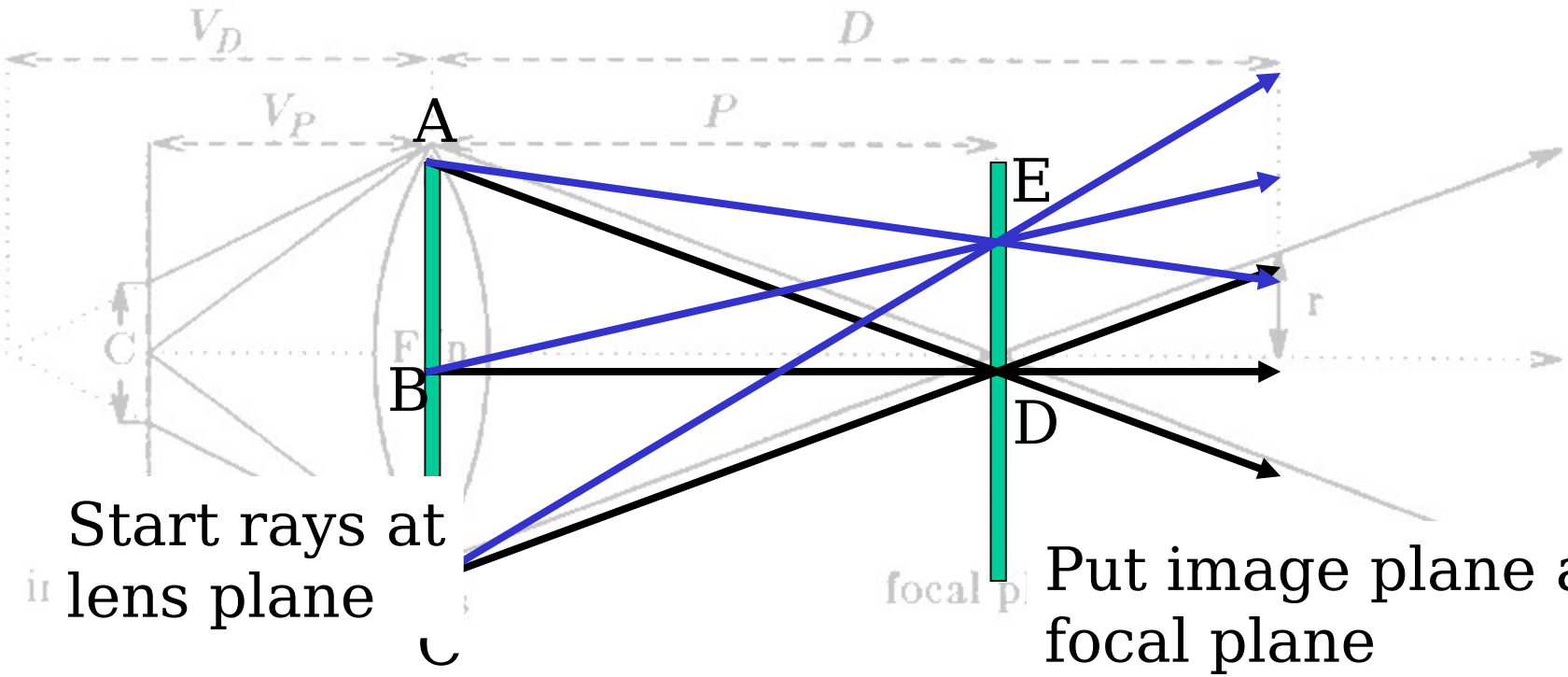
$C = (|V_D - V_P|/V_D)\ (F/n)$

$r = ½\ (F/n)\ (D\text{-}P)/P$

$R = (-V_P/D)\ r$

$R = ½\ C$

Implementation

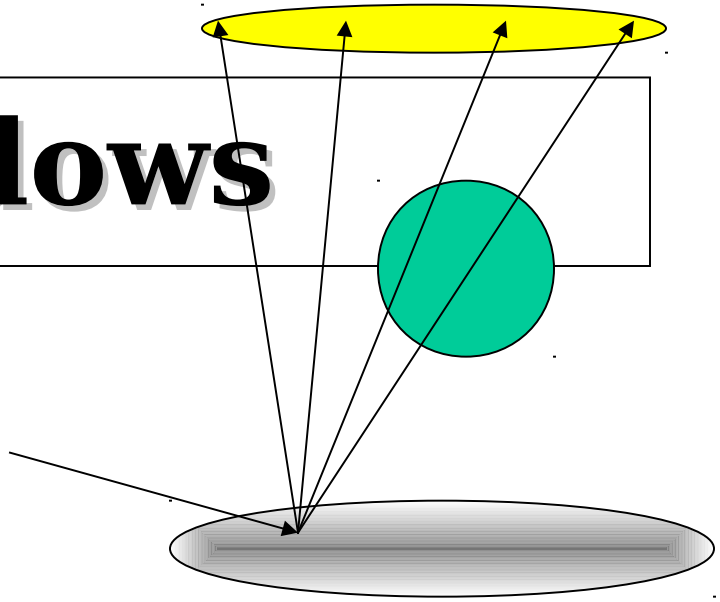


Standard ray tracing:
Pixel D uses ray BD
Pixel E uses ray BE
All rays emanate from B

Distributed ray tracing:
Pixel D uses rays AD, BD, CD
Pixel E uses rays AE, BE, CE
Rays emanate from lens plan

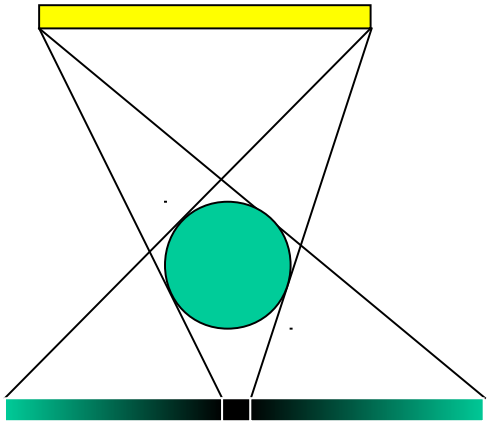
Play Video Clip

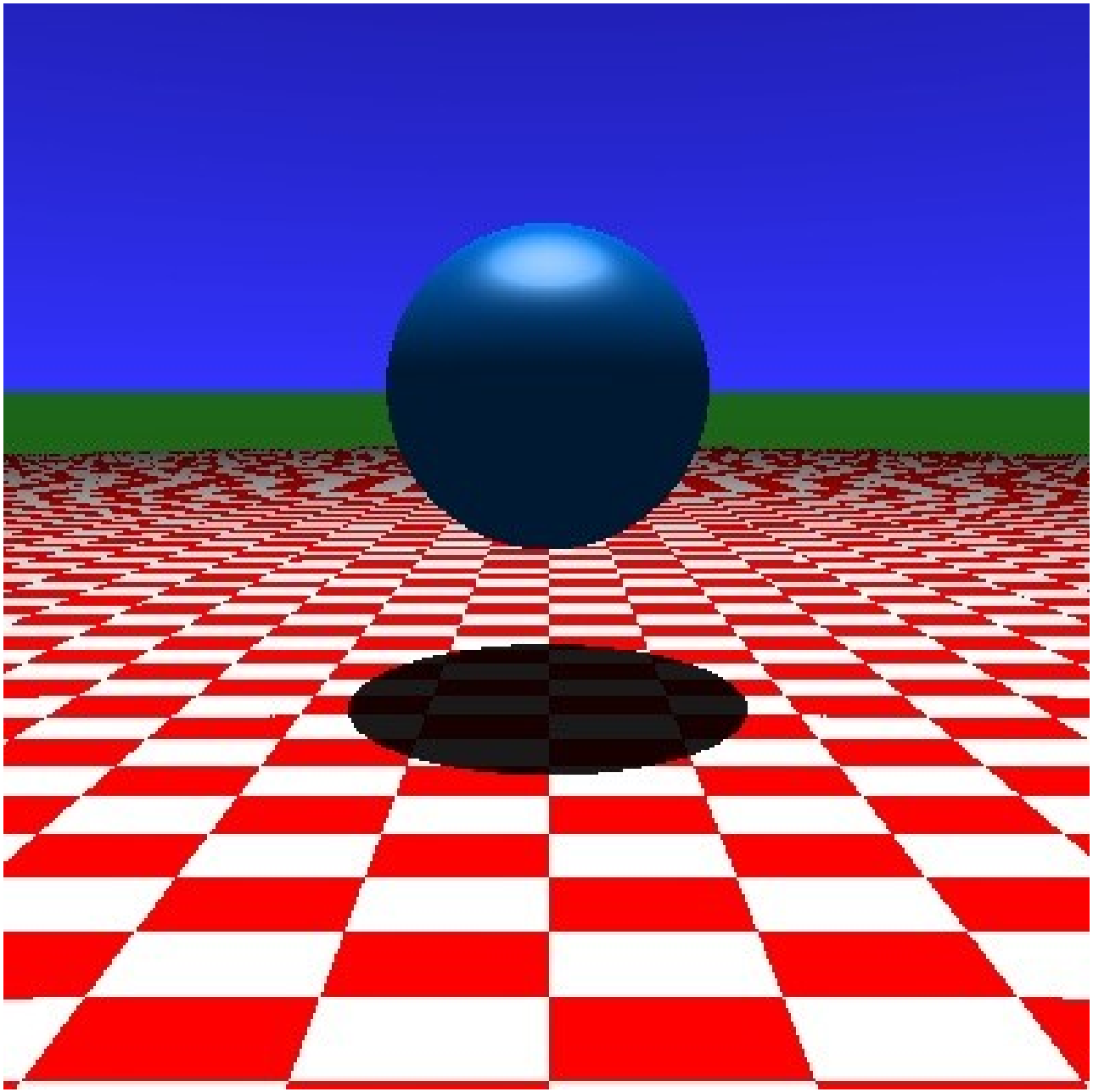
Soft Shadows



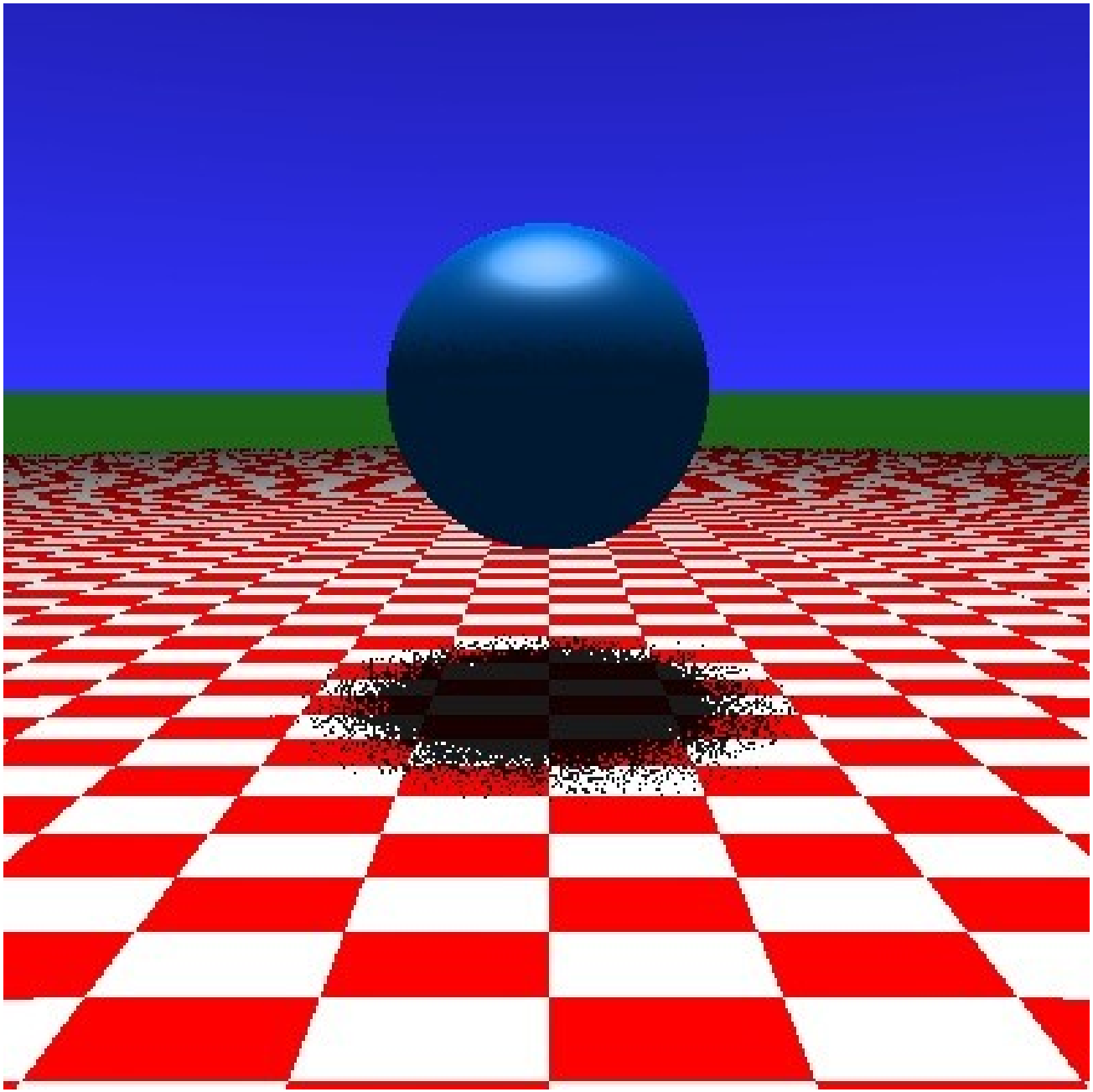
50% illuminated

- Point light sources unrealistic
- Use area light source
- Cast shadow rays from surface to different locations on light
- Hits/rays = % illuminated
- Jitter locations to remove aliasing artifacts

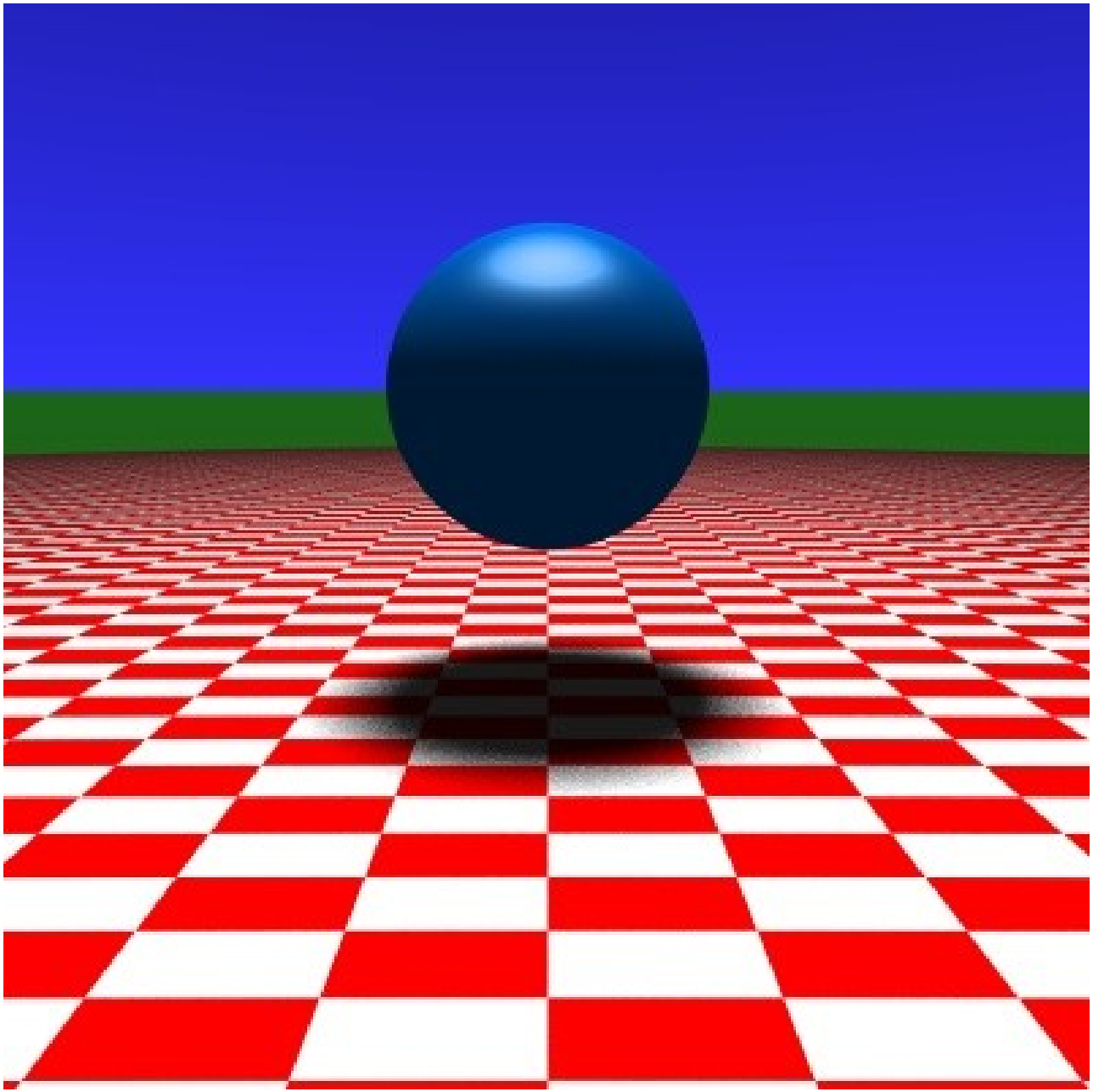




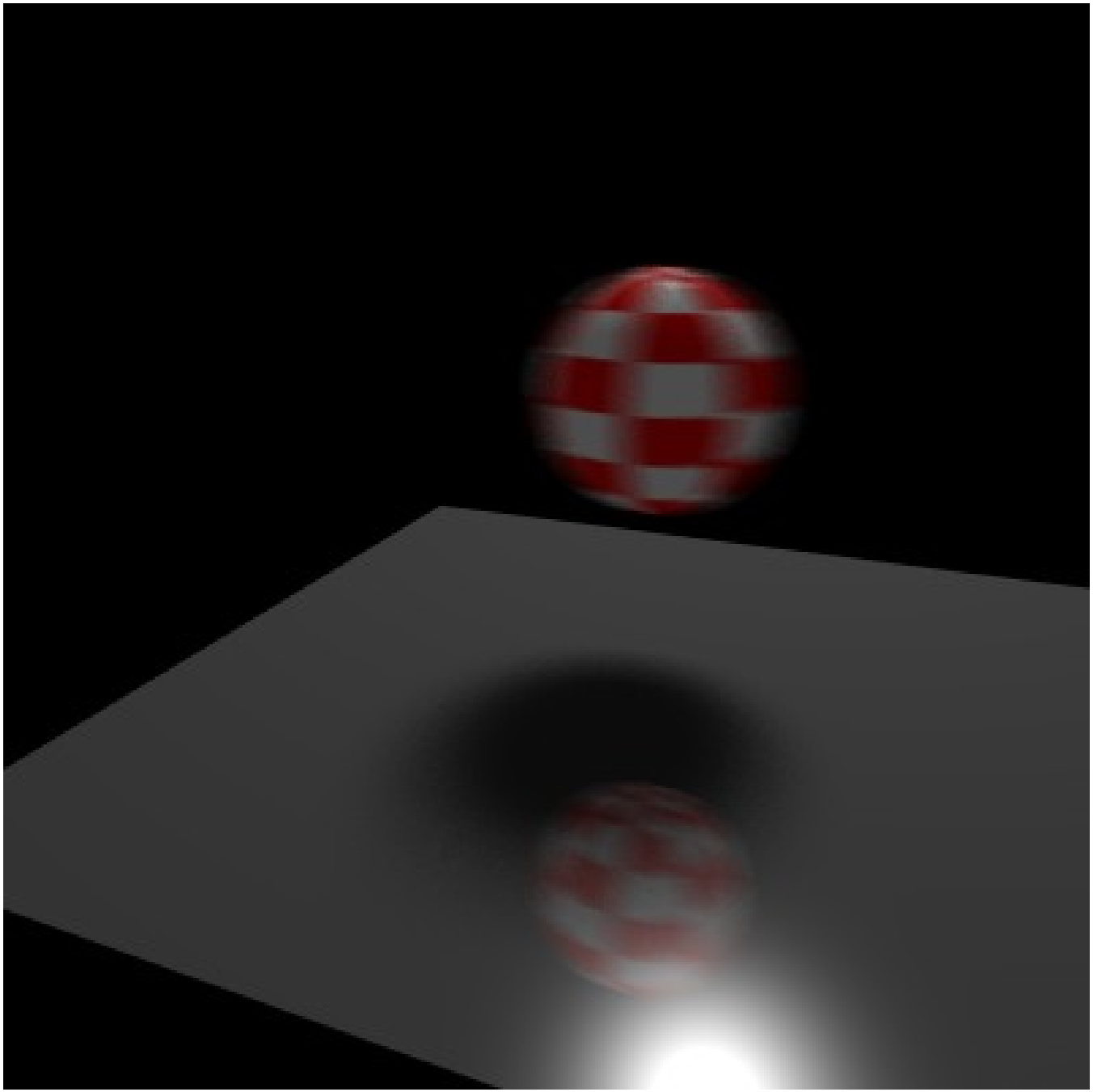
Point light source above the object



Area light source, 1 sample per pixel

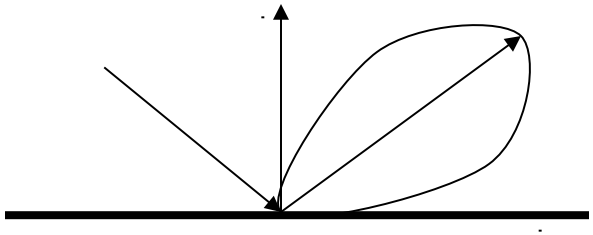


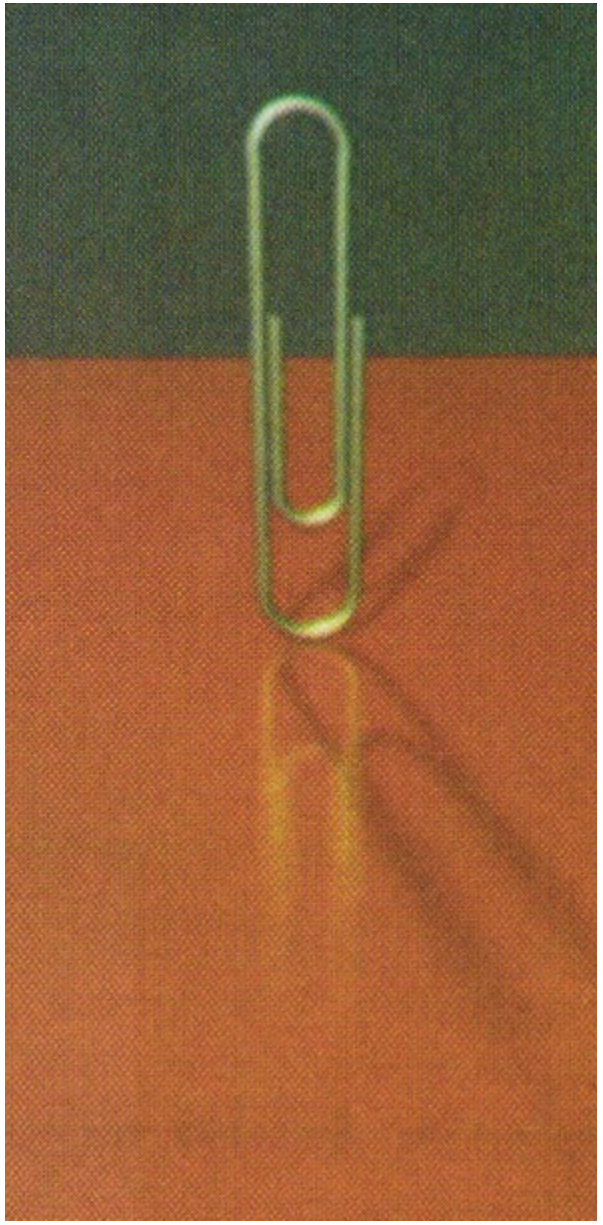
Area light source, 256 samples per pixel

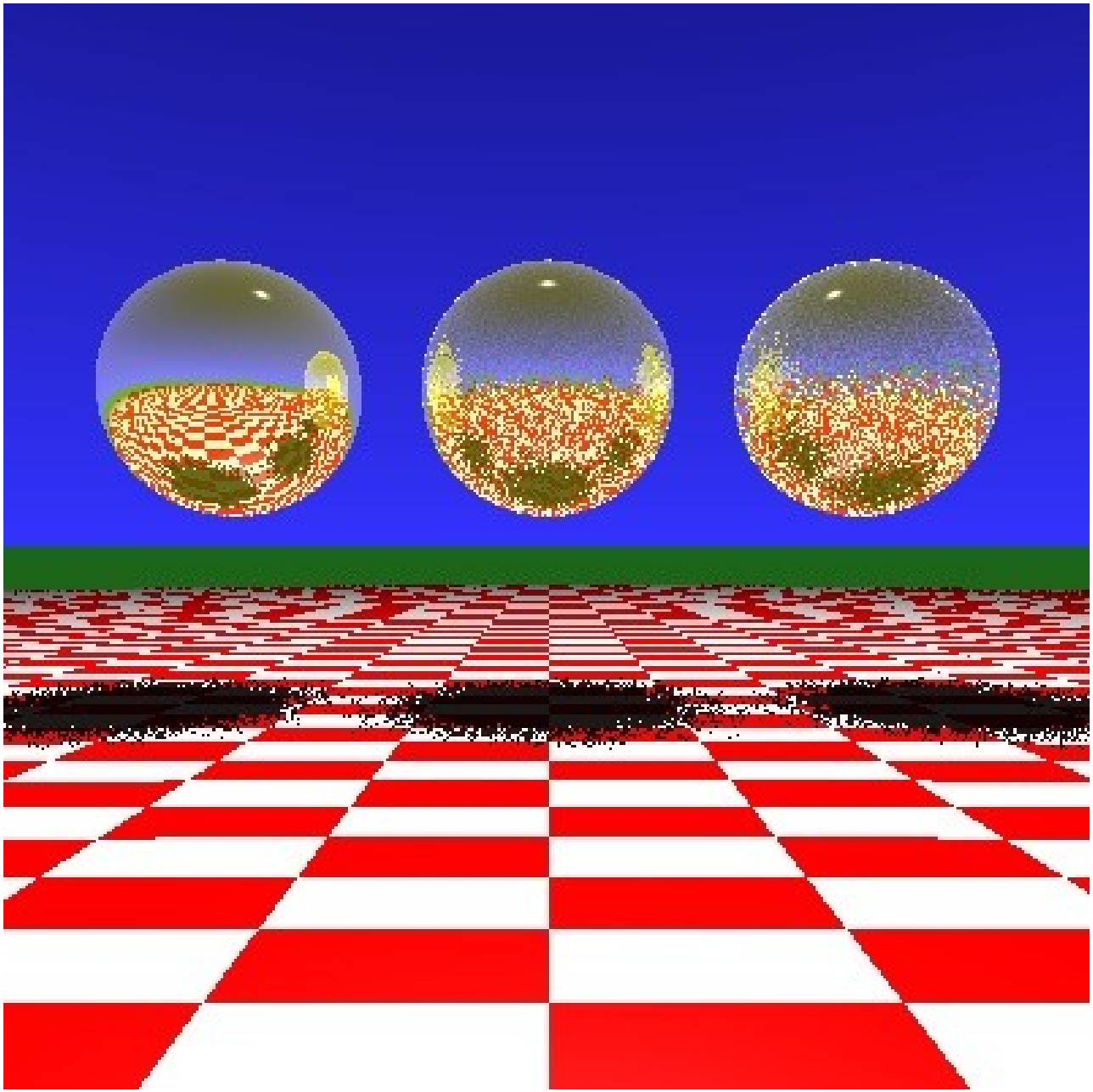


Glossy Surfaces

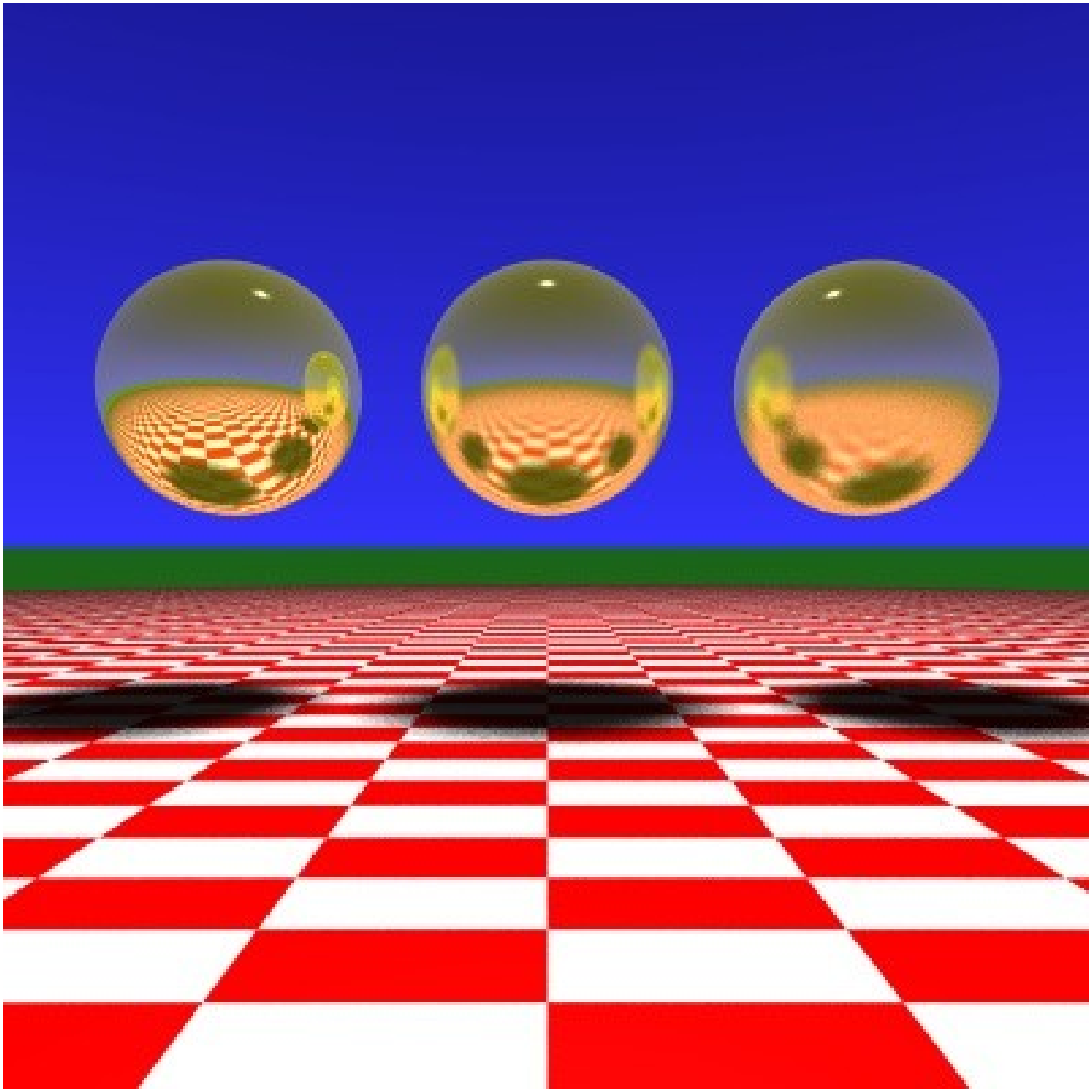
- Surface microfacets perturb reflection rays
- Nearby objects reflect more clearly than distant objects
- Perturb reflection ray direction
- Reconstruct using appropriate filter
  - Strongest contribution in reflected direction
  - Falls off nearby



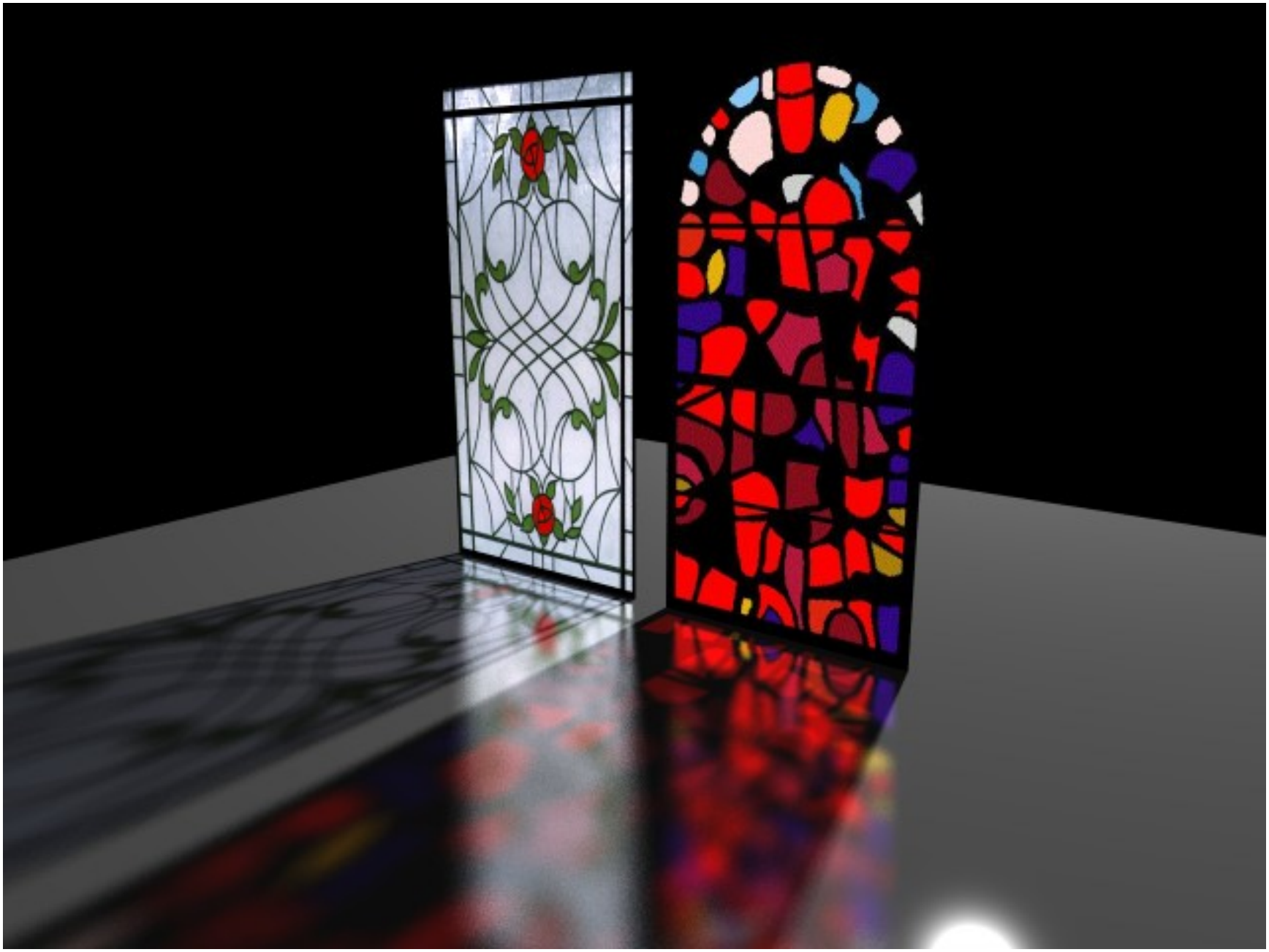




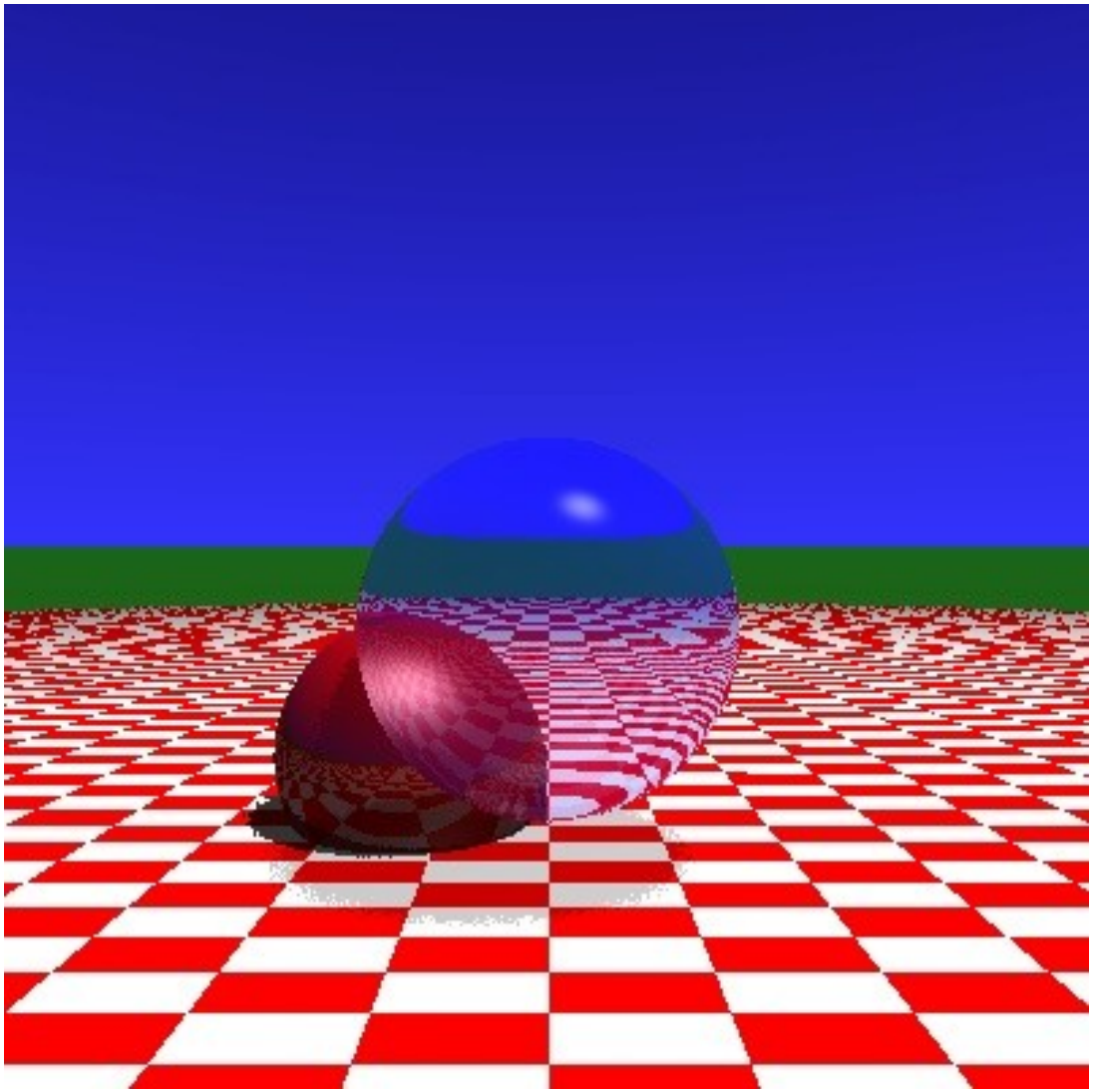
1 sample per pixel



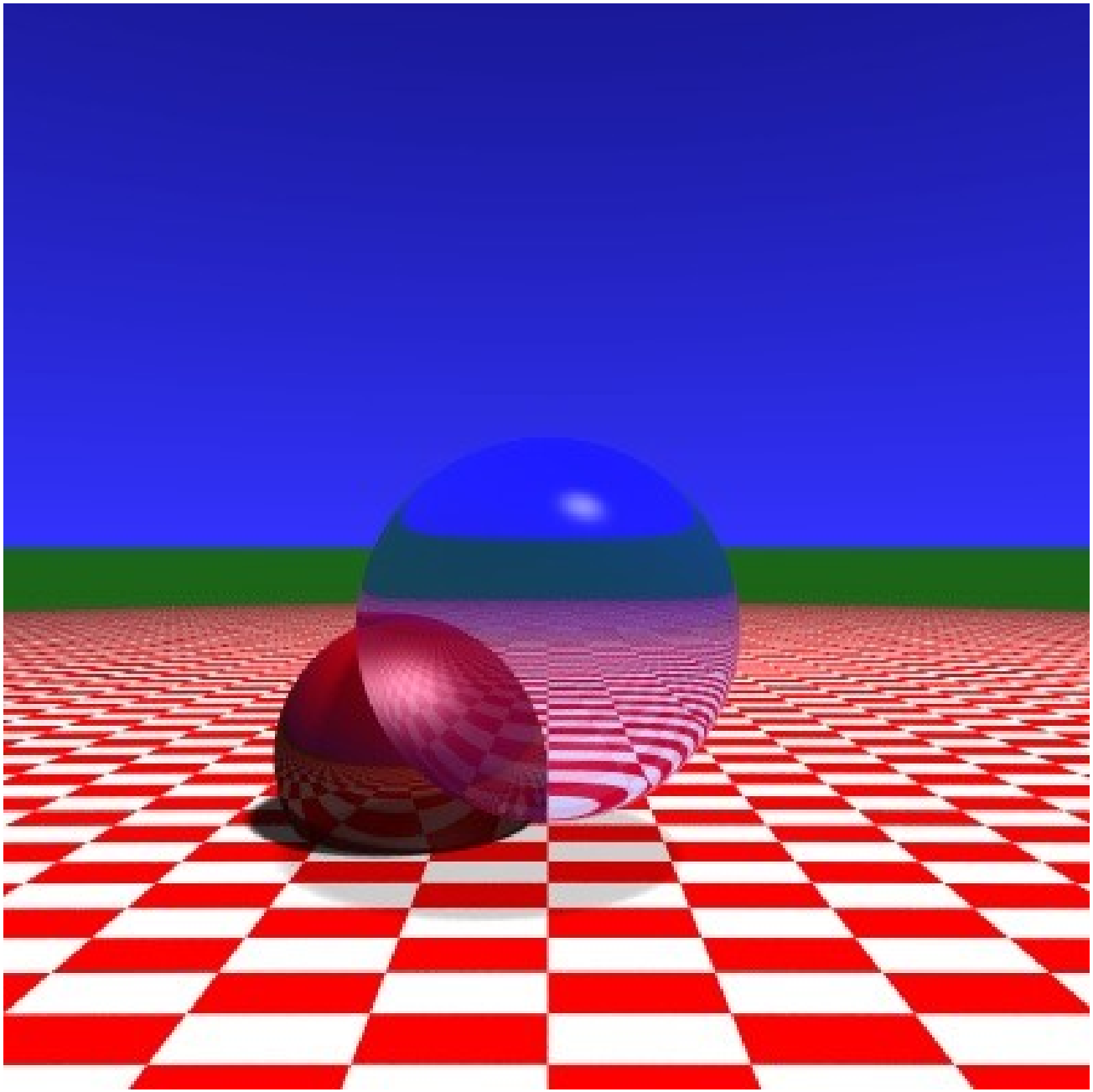
256 samples per pixel



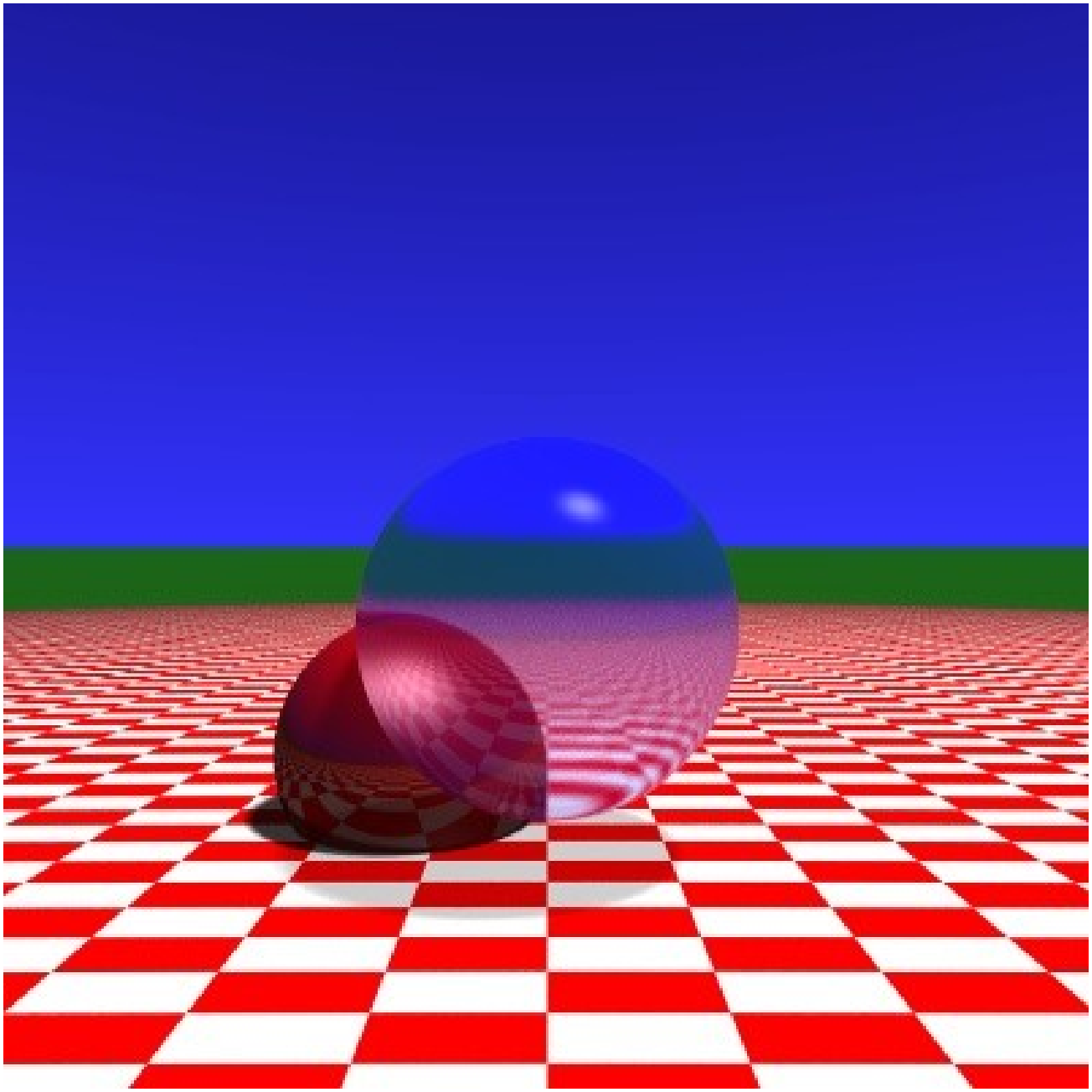
Translucency



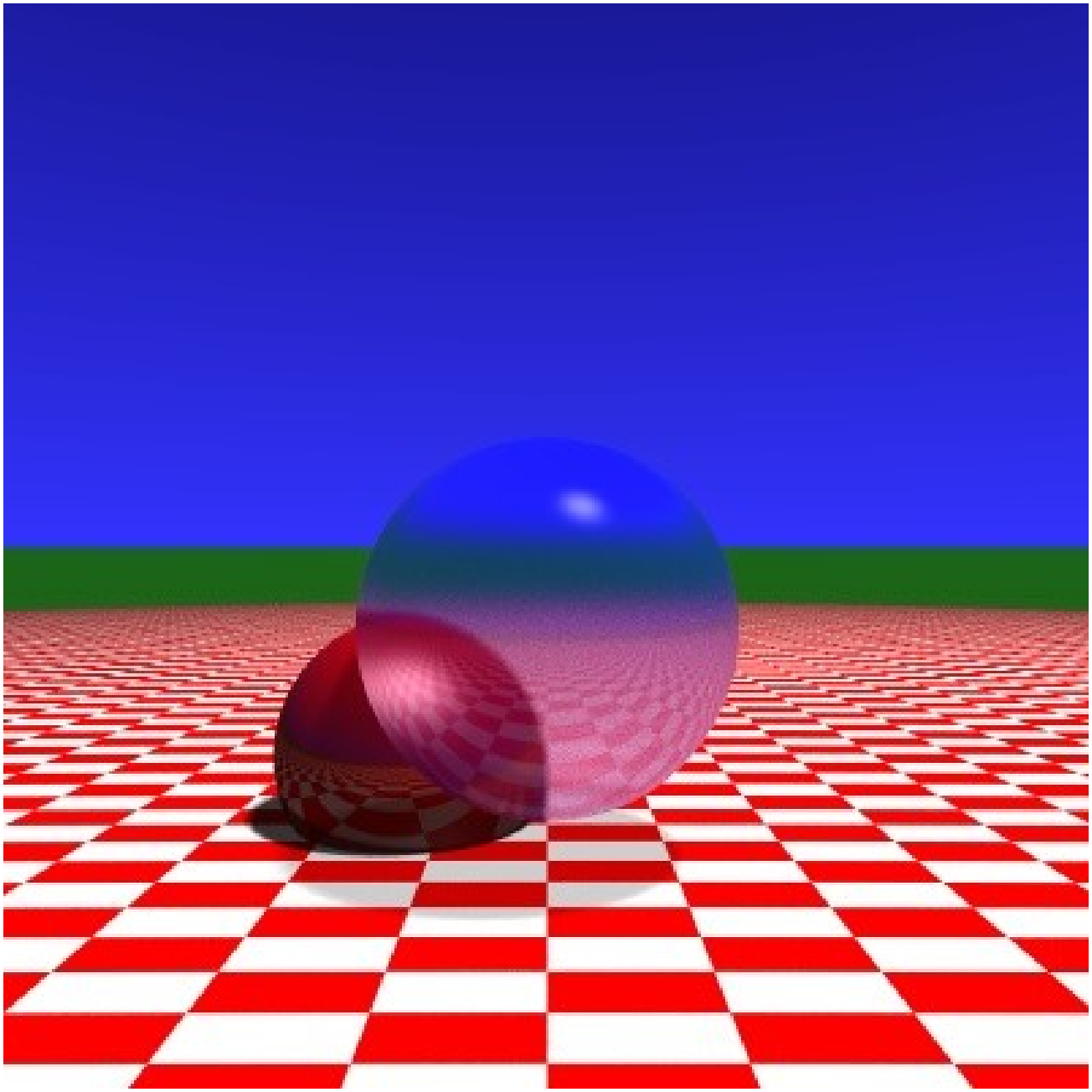
1 sample per pixel



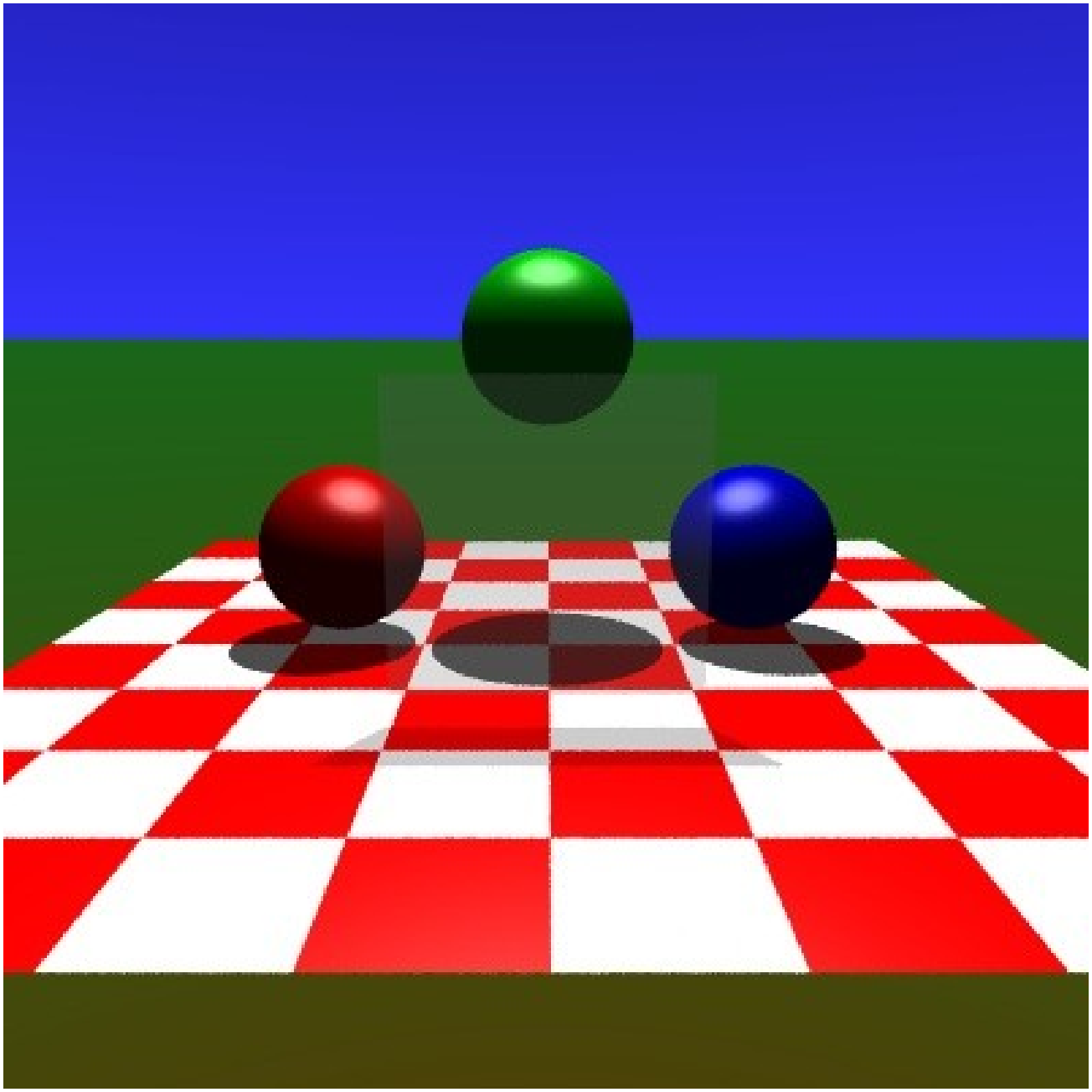
256 samples per pixel



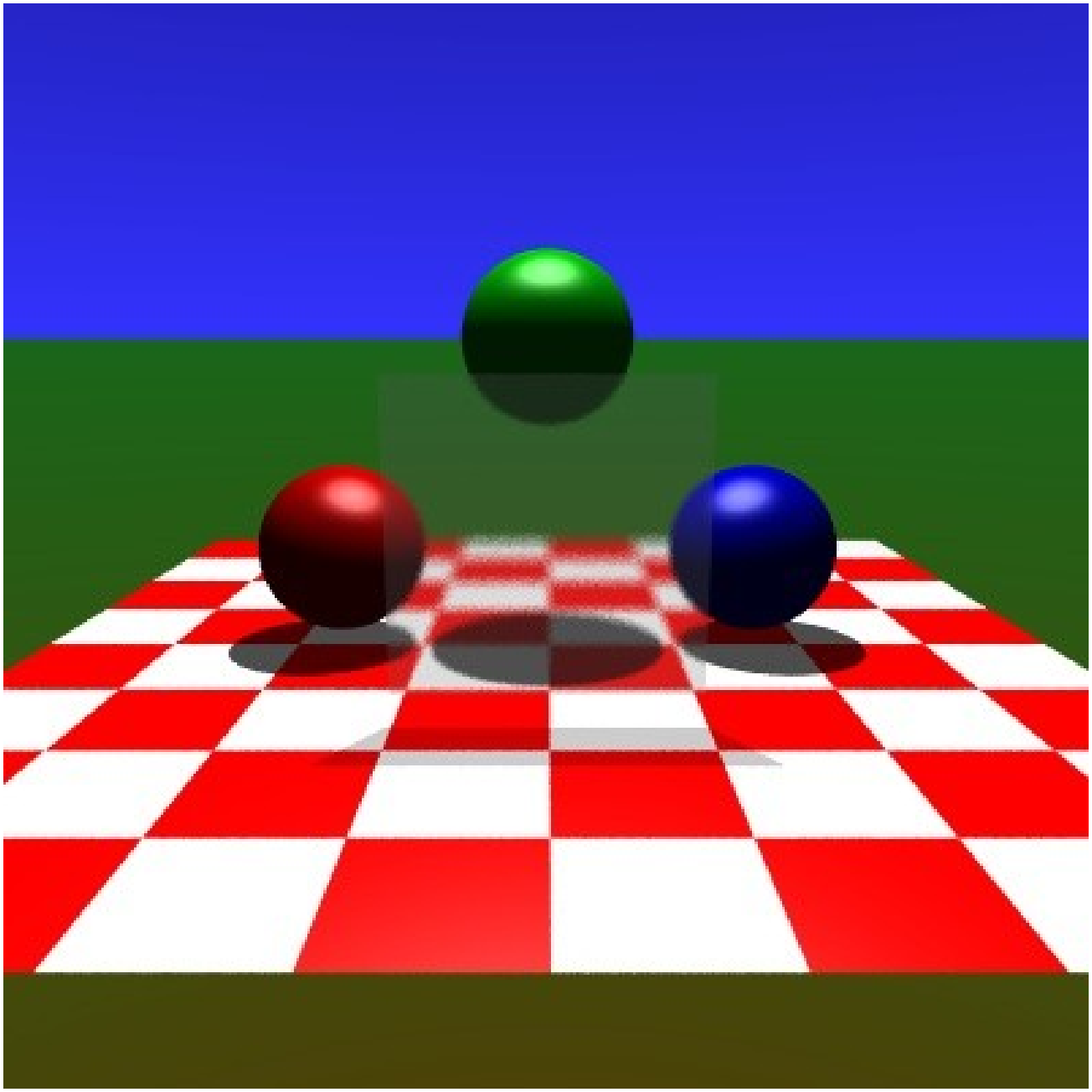
Decreased Gloss, 256 samples per pixel



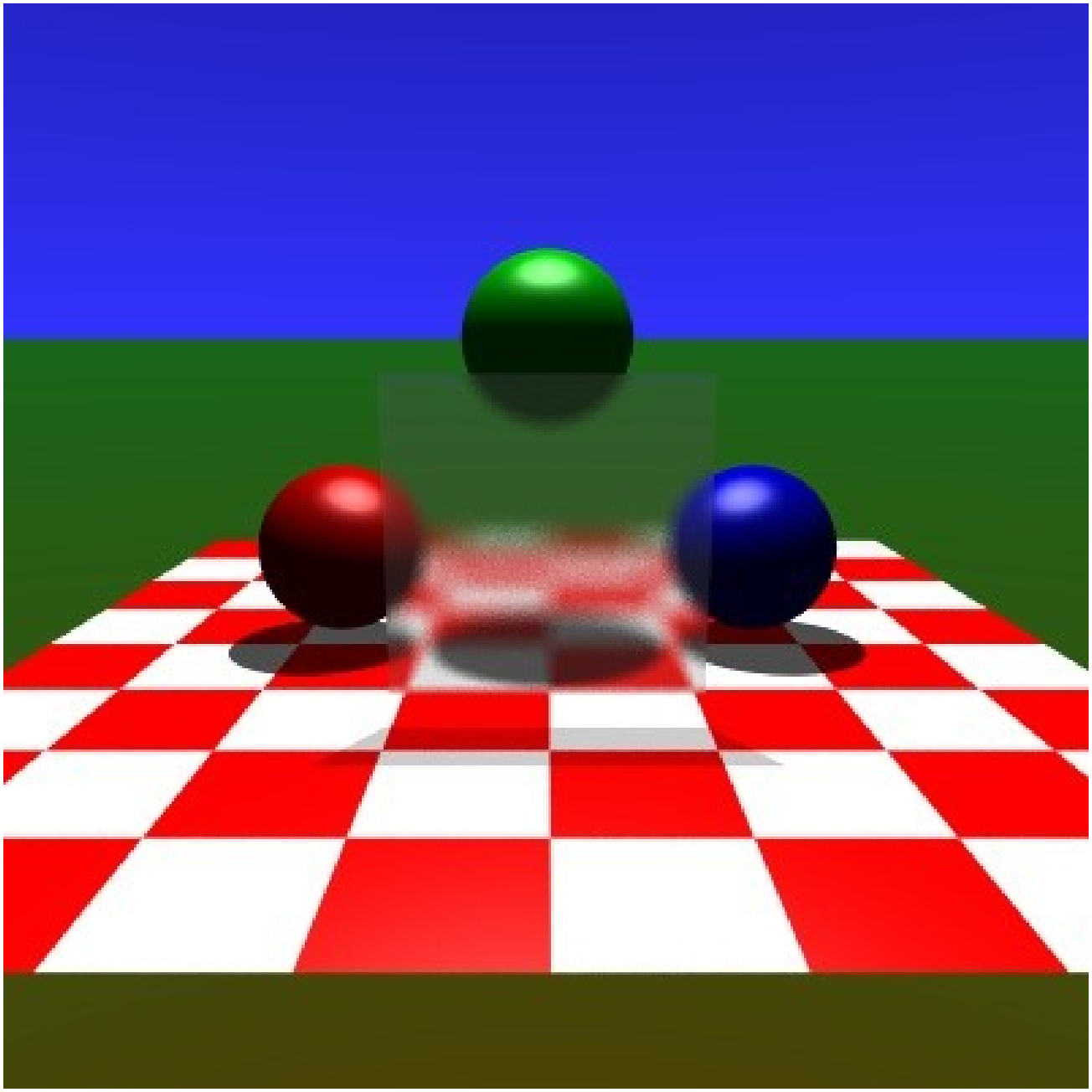
Decreased Gloss, 256 samples per pixel



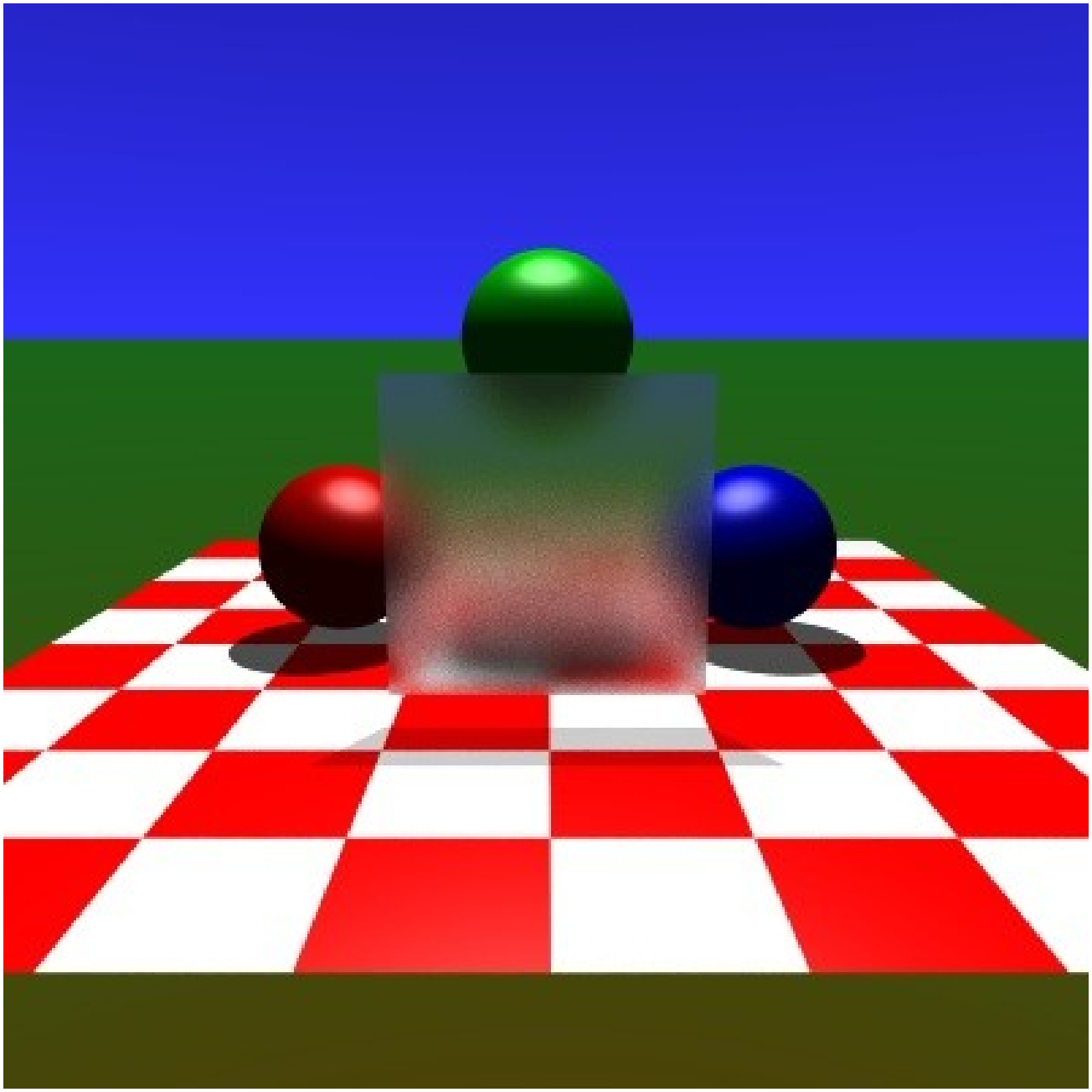
Colorless Square Perfectly Shiny



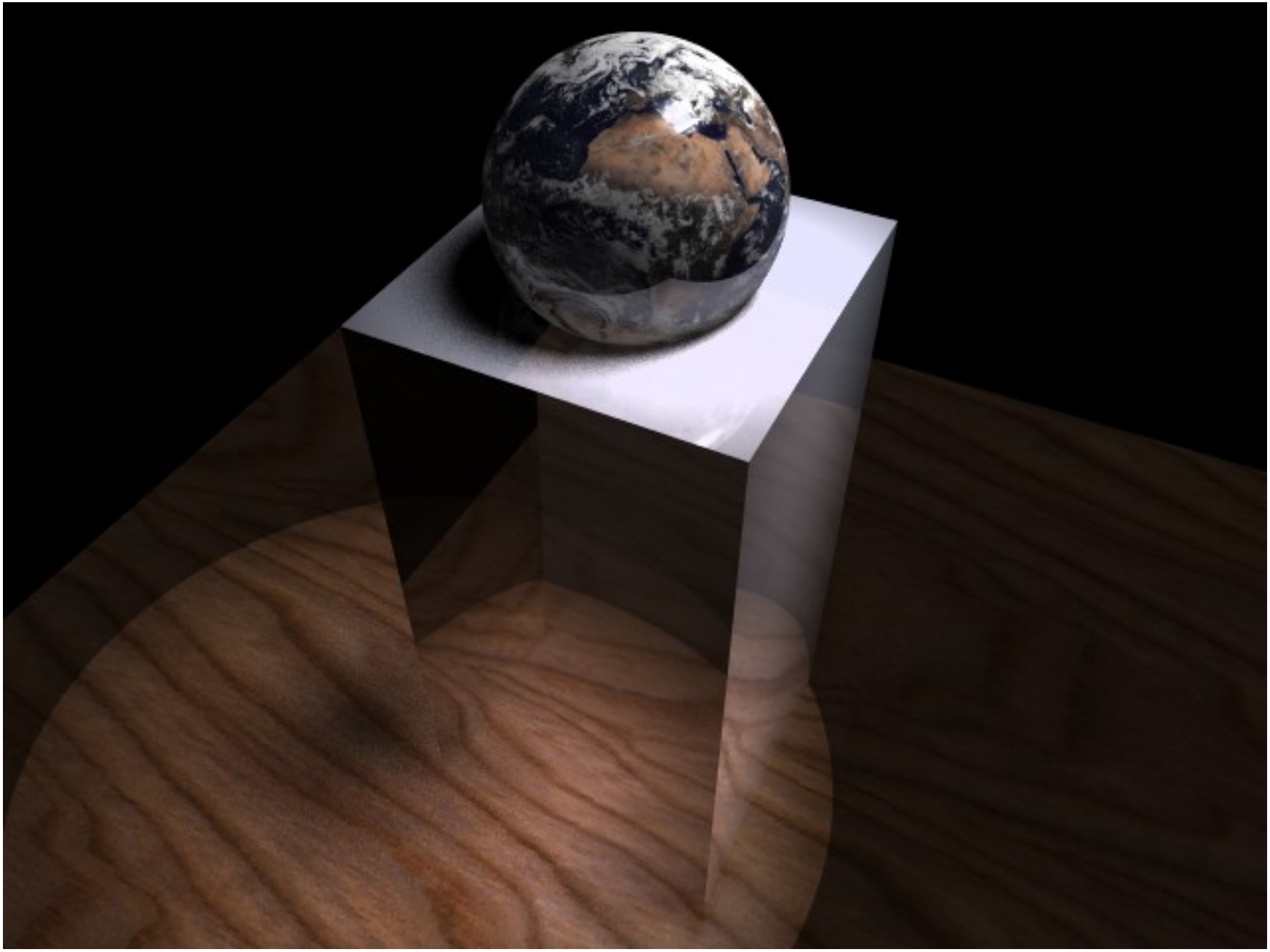
Colorless Square Decreased Shininess



Colorless Square Decreased Shininess



Frosted Glass



So what's the problem?

Casting *n* rays through each pixels takes *n* times as long

- Solution:
  - Trace pixel rays
  - Analyze variance
    - If variance is too high
      - Add more samples
      - Reconstruct new results
  - Repeat

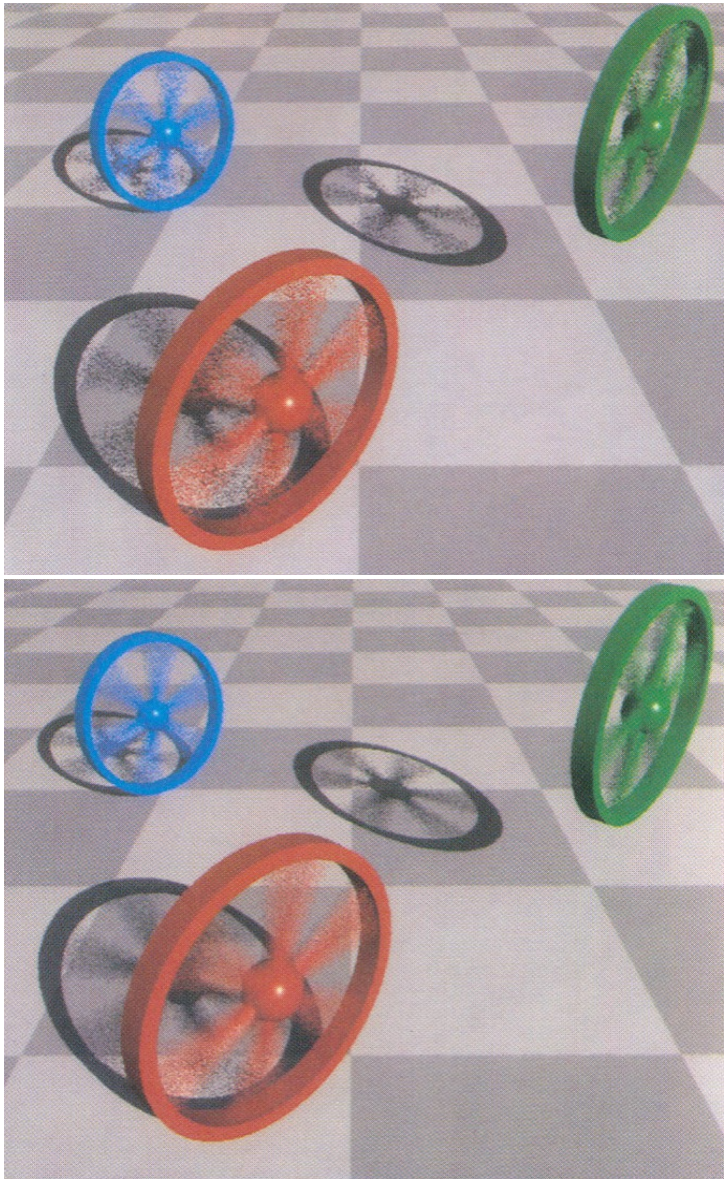
Time to Crunch

There are several approaches to speeding up computations:

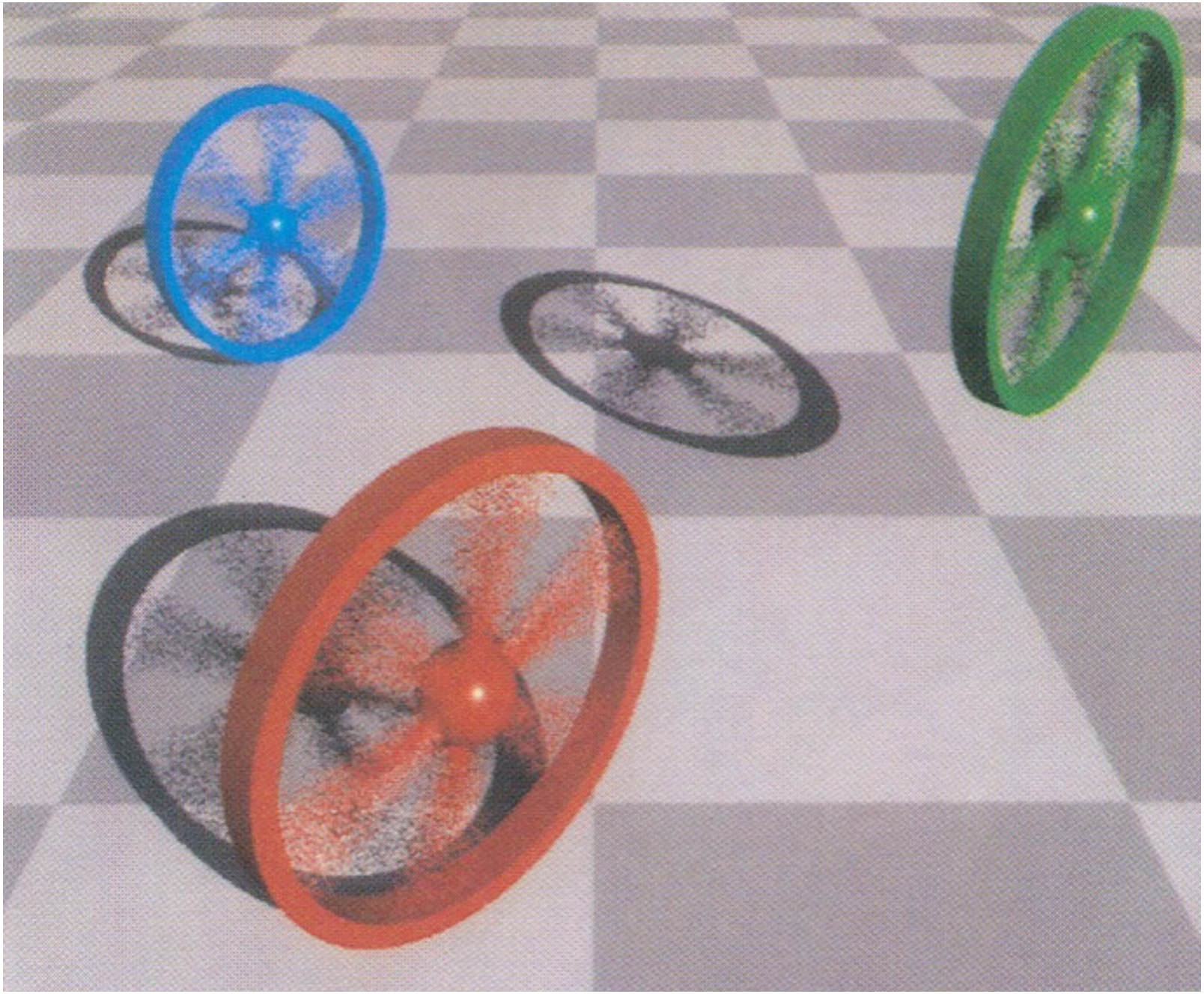
1. Use faster machines
2. Use specialized hardware, especially parallel processors.
3. Speed up computations by using more efficient algorithms
4. Reduce the number of ray - object computations
    - Adaptive depth control
    - Bounding Volumes
    - First - hit Speedup

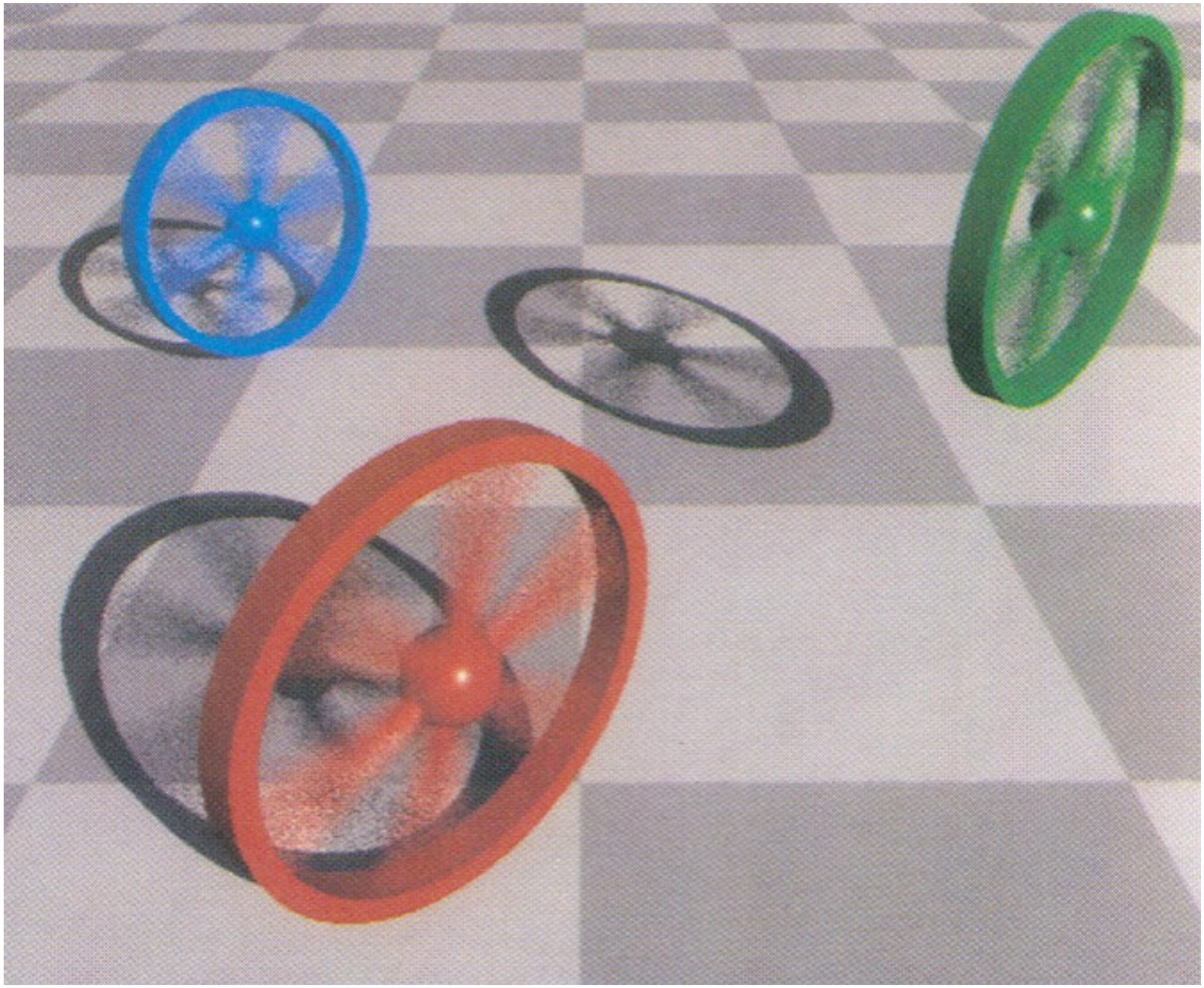
Distributed Sampling

- Supersampled antialiasing
  - Jittering removed jaggies, Moires
  - Ordered jitter, Poisson balanced distribution
- Decorrelation inhibits aliasing
- Need to decorrelate samples
  - across all parameter axes
  - between all parameter axes



?





Advanced Camera Models